UNIVERSIDAD CARLOS III DE MADRID
ESCUELA POLITÉCNICA SUPERIOR

PROYECTO DE FIN DE CARRERA
INGENIERÍA SUPERIOR EN INFORMÁTICA

Sistema de Registro Telemático Basado en una Arquitectura Orientada a Servicios

Autor **Bernardo Cabaleiro Barciela**
Tutor **Juan Miguel Gómez Berbís**

Agradecimientos

Me gustaría en primer lugar agradecer a Juanmi todo el apoyo que me ha dado, tanto con el proyecto como fuera de él.

También a los compañeros de Everis, dónde comenzó a gestarse este trabajo.

A mis amigos de Vigo, que aunque no los veo todo lo que me gustaría, ellos saben lo importantes que son para mí.

También quiero acordarme de "Los Cebollitas", grandes deportistas y mejores personas. Han sido mis compañeros y mis amigos en mi aventura en Madrid, y sin ellos nada habría sido lo mismo.

A mi familia, que me ha apoyado en todo momento, se han preocupado por mí y me han ayudado más que nadie.

A mi hermano Lucas, a modo de disculpa, porque aunque él no lo quiera le toca tener que aguantarme muchos años más.

Por último, quiero agradecer muy especialmente a mi padre, porque no se cansó de cantar todos los goles desde el principio de la carrera, y a mi madre, porque sí, porque es mi madre favorita. Aunque suene a tópico no podría ser más cierto, sin ellos no habría sido posible.

A todos:
¡Muchas gracias!

ÍNDICE DE CONTENIDOS

ÍNDICE DE TABLAS

ÍNDICE DE ILUSTRACIONES

1. INTRODUCCIÓN

La Ley 11/2007, de 22 de junio, de acceso electrónico de los ciudadanos a los Servicios Públicos otorga a los ciudadanos el **derecho a relacionarse con la administración pública** por **medios electrónicos**. Por ello es necesario crear diferentes aplicaciones, adaptadas a cada una de las administraciones, incluyendo generales, autónomas y locales, que cumplan con el cometido deseado.

Este servicio tendrá la misma **validez legal** que el servicio prestado por el registro presencial, y realizará las mismas funciones, pero se empleará el **canal telemático**.

De esta manera se pretende agilizar las relaciones entre administración y usuarios, teniendo repercusiones positivas en ambas:

- Para el ciudadano:
  - Evita desplazamientos innecesarios.
  - Amplia el horario de atención por parte de la administración.
  - Reduce el tiempo empleado en realizar un trámite.

- Para la administración:
  - Reduce la masificación de la atención al público.
  - Como consecuencia de lo anterior, aumenta la productividad de los empleados.
  - Disminuye el tiempo de respuesta frente a cualquier trámite.
  - Aumentar la satisfacción de los usuarios, y por tanto la valoración positiva de la administración.
  - Posibilita el ofrecimiento de nuevos servicios.

Cabe destacar que parte de los servicios ofrecidos son comunes a la gran parte de las administraciones, por lo que se antoja necesaria una arquitectura reutilizable que simplifique el desarrollo y abarate los costes.

Una de estas partes comunes, quizá una de las más importantes, será el **registro y la validación de los usuarios**. Esta parte representa el punto de entrada formal de todas las comunicaciones que el ciudadano desee realizar, diferenciando cada una de ellas. Además, en cada interacción con el usuario, es imprescindible registrar cuál es la información recibida por la aplicación, así como generar un justificante apropiado que certifique que el trámite se ha producido y que es válido. De esta manera, tanto administración como usuario tienen constancia formal del trámite.

Por otra parte, hay que garantizar que la información transmitida no esté manipulada, por lo que tanto las solicitudes del usuario como los justificantes correspondientes deberán contener una **firma digital**.

Por último, los trámites pueden necesitar documentos adjuntos, que deberán ser enviados por el usuario y recibidos por la aplicación, momento en el cual serán almacenados y catalogados de manera segura, lo que incluye asegurar su disponibilidad, integridad y confidencialidad.

El servicio que debe prestar el Registro Telemático se encuentra determinado por el Real Decreto 209/2003, de 21 de febrero, por el que se regulan los registros y las notificaciones telemáticas, así como la utilización de medios telemáticos para la sustitución de la aportación de certificados por los ciudadanos.

1.1 Objetivos

Durante el presente proyecto se desarrollará un **servicio web** que proporcione a las aplicaciones de la administración pública la funcionalidad de **registrar telemáticamente** a los usuarios.

Este servicio está diseñado para encajar dentro de un sistema mayor. Este sistema estaría basado en una **arquitectura orientada a servicios** o SOA (Service Oriented Architecture), en la que, tal y como su nombre indica, se dividirán las funcionalidades en distintos servicios web. Estos servicios estarán comunicados entre sí por una estructura ESB (Enterprise Service Bus). Esta estructura consiste en un bus de datos por el que se transmiten todos mensajes que comunican los diferentes servicios, de manera que conecta las diferentes partes del sistema.

Esto permite que la aplicación sea fácilmente ampliable, ya que posibilita que se añadan nuevas funcionalidades de manera modular. También se consigue que la aplicación tenga un mantenimiento sencillo, ya que en caso de que un servicio quede obsoleto puede ser sustituido por uno nuevo sin afectar al resto de componentes.

Asimismo, este servicio podrá utilizarse como un conector con registros telemáticos externos, es decir, será capaz de comunicarse con aplicaciones de terceros que sustituirán la propia herramienta. De esta manera, se consigue que las administraciones que dispongan de una aplicación propia para este fin puedan integrarla en el nuevo sistema.

Complementariamente, se implementará un servicio que dispondrá de la funcionalidad de conector con servicios de **firma digital** externos, necesarios para el funcionamiento del registro.

Por otra parte, uno de los objetivos principales en este desarrollo es crear una aplicación flexible, de manera que pueda **adaptarse a cualquier sistema de la administración pública**, lo que diferenciará este producto de otros disponibles en el mercado.

Para aumentar esta flexibilidad se empleará un lenguaje de programación que disponga de una amplia variedad de **herramientas de libre distribución**. Entre estas, se seleccionarán aquellas más apropiadas para este proyecto y se comprobará si proveen de la funcionalidad requerida.

1.2 Estructura del Documento

Esta memoria está dividida en las siguientes secciones:

- **Estado del Arte**: En este apartado se comenta la situación actual del problema a tratar, así como las soluciones tecnológicas existentes y las herramientas software que se van a emplear en el proyecto.
- **Modelo**: En el modelo se realizará un análisis sobre la aplicación a desarrollar, especificando su funcionamiento y capacidades.
- **Arquitectura e Implementación**: En este apartado se detallará a bajo nivel cómo está implementada la aplicación.
- **Conclusiones**: En el apartado de conclusiones se explicarán los resultados del proyecto y se realizará una valoración final de la aplicación.
- **Líneas Futuras**: Para terminar se expondrán algunas sugerencias sobre los caminos que podría seguir el desarrollo de la aplicación.

De esta manera se pretende respetar la secuencia cronológica de realización del proyecto, ya que comienza con el proceso de documentación necesario para abordarlo, continúa con el análisis y el diseño del propio proyecto, y finaliza con los resultados obtenidos tras la implementación.

1.3 Definiciones y Siglas

1.3.1 Definiciones

- Asiento registral: Un asiento registral es un conjunto de datos que forman una anotación en el registro.
- Aplicación usuaria: En este contexto, se denomina aplicación usuaria al programa que emplea el ciudadano, y por lo tanto es el que desencadena el funcionamiento de los servicios.
- Certificado digital: Estructura de datos que contiene la clave pública de un usuario junto con sus datos de identificación. Nombre y DNI. Los emiten los Prestadores de Servicios de Certificación y sirven para verificar las firmas e identificar al firmante.
- Custodia documental: La custodia documental es un proceso mediante el cual se almacenan documentos electrónicos, de manera que éstos sólo pueden ser accedidos por las personas autorizadas.
- Firma Digital: Operación criptográfica realizada sobre un documento con la clave privada de su único poseedor.
- Huella digital: Se denomina huella digital al resultado de aplicar un algoritmo de resumen a un documento digital. A partir de esta huella no es posible obtener el documento original, pero puede comprobarse
- Servicio restringido: Tipo de servicio al que sólo pueden acceder las personas o aplicaciones autorizadas.
- Tag (XML): Cada una de las etiquetas que marcan los principios y finales de bloque en la estructura de un fichero XML.
- Ticket: Se denomina ticket a una cadena de caracteres que identifica a una aplicación.

1.3.2 Siglas

- ACA: **A**utoridad de **C**ertificación de la **A**bogacía
- ACCV: **A**utoridad de **C**ertificación de la **C**omunidad **V**alenciana
- ANCERT: **A**gencia **N**otarial de **Cert**ificación
- ANF: **A**sociación **N**acional de **F**abricantes
- API: **A**pplication **P**rogramming **I**nterface (Interfaz de programación de aplicaciones)

- CAdES: **C**MS **Ad**vanced **E**lectronic **S**ignatures (Firma electrónica avanzada CMS)
- CSS: **C**ascade **S**tyle **S**heet (Hojas de estilo en cascada)
- CMS: **C**ryptographic **M**essage **S**yntax (Sintaxis de mensaje criptográfico)
- DAO: **D**ata **A**ccess **O**bject (Objeto de acceso a datos)
- DCOM: **D**istributed **C**omponent **O**bject **M**odel (Modelo de objetos de componentes distribuidos)
- DTD: **D**ocument **T**ype **D**efinition (Definición de tipo de documento)
- DOM: **D**ocument **O**bject **M**odel (Modelo de objetos de documento)
- E/S: **E**ntrada / **S**alida
- EJB: **E**nterprise **J**ava**B**ean
- FNMT: **F**ábrica **N**acional de **M**oneda y **T**imbre
- GNU- LGPL: **GNU** **L**esser **G**eneral **P**ublic **L**icense (Licencia pública general reducida de GNU)
- HTTP: **H**yper**t**ext **T**ransfer **P**rotocol (Protocolo de transferencia de hipertexto)
- HTTPS: **H**yper**t**ext **T**ransfer **P**rotocol **S**ecure (Protocolo seguro de transferencia de hipertexto)
- HQL: **H**ibernate **Q**uery **L**anguage (Lenguaje de consultas de Hibernate)
- IETF: **I**nternet **E**ngineering **T**ask **F**orce (Grupo de tareas de ingeniería de internet)
- JDBC: **J**ava **D**ata**B**ase **C**onnectivity
- JPA: **J**ava **P**ersistence **A**PI (API de persistencia de Java)
- JVM: **J**ava **V**irtual **M**achine (Máquina virtual de Java)
- J2EE: **J**ava **2** Platform, **E**nterprise **E**dition
- MD2: **M**essage-**D**igest Algorithm **2** (Algoritmo de resumen del mensaje 2)
- MD5: **M**essage-**D**igest Algorithm **5** (Algoritmo de resumen del mensaje 5)
- PKCS: **P**ublic-**K**ey **C**ryptography **S**tandars (Estándar de criptografía de clave pública)
- POJO: **P**lain **O**ld **J**ava **O**bject
- RCP: **R**emote **P**rocedure **C**alling (Llamada a procedimiento remoto)
- RFC: **R**equest **F**or **C**omments (Petición de comentarios)

- SAX: **S**imple **A**PI for **X**ML (API simple para XML)
- SCR: **S**ervicio de **C**ertificación de los **R**egistradores
- SHA: **S**ecure **H**ash **A**lgorithm (Algoritmo de Hash Seguro)
- SOA: **S**ervice **O**riented **A**rchitecture (Arquitectura orientada a servicios)
- SOAP: **S**imple **O**bject **A**ccess **P**rotocol (Protocolo simple de acceso a objetos)
- SQL: **S**tructured **Q**uery **L**anguage (Lenguaje de consulta estructurado)
- StAX: **St**reaming **A**PI for **X**ML
- TIC: **T**ecnologías de la **I**nformación y **C**omunicación
- UDP: **U**ser **D**atagram **P**rotocol (Protocolo de datagrama de usuario)
- URI: **U**niform **R**esource **I**dentification (Identificador uniforme de recursos)
- URL: **U**niform **R**esource **L**ocator (Localizador uniforme de recursos)
- WSDL: **W**eb **S**ervices **D**escription **L**anguage (Lenguaje de descripción de servicios web)
- XAdES: **X**ML **Ad**vanced **E**lectronic **S**ignatures (Firma electrónica avanzada XML)
- XML: E**x**tended **M**arkup **L**anguage (Lenguaje de marcado extensible)
- XMLdSig: **XML D**igital **Sig**nature (Firma digital de XML)

Planificación

El proyecto da comienzo el día 13 de Octubre de 2008. Se estima una media de cuatro horas de trabajo por día, por lo que el fin del proyecto se calculó para el 8 de Junio de 2009.

La planificación del desarrollo del proyecto se representa, con los plazos detallados, en el siguiente diagrama:

Id	Nombre de tarea	Duración
1	**Estudio del entorno**	**7 días**
2	Análisis de registros telemáticos	2 días
3	Análisis de aplicaciones de firma digital	2 días
4	Análisis de alternativas de software libre	3 días
5	**Análisis**	**12 días**
6	Análisis de Registro Telemático	6 días
7	Análisis de Firma Electrónica	6 días
8	**Diseño**	**18 días**
9	Diseño de Registro Telemático	9 días
10	Diseño de Firma Electrónica	9 días
11	**Registro Telemático**	**65 días**
12	Desarrollo	20 días
13	Pruebas	40 días
14	Documentación	5 días
15	**Firma Electrónica**	**45 días**
16	Desarrollo	15 días
17	Pruebas	25 días
18	Documentación	5 días
19	Pruebas de integración	10 días
20	Documentación Proyecto Fin de Carrera	26 días

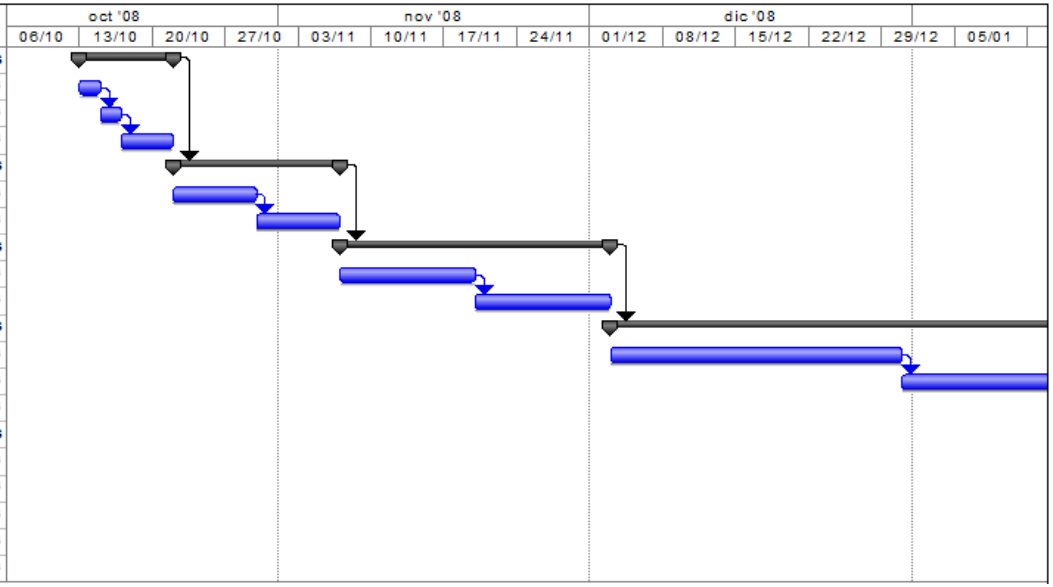

Ilustración 1. Diagrama de Gantt (Parte 1)

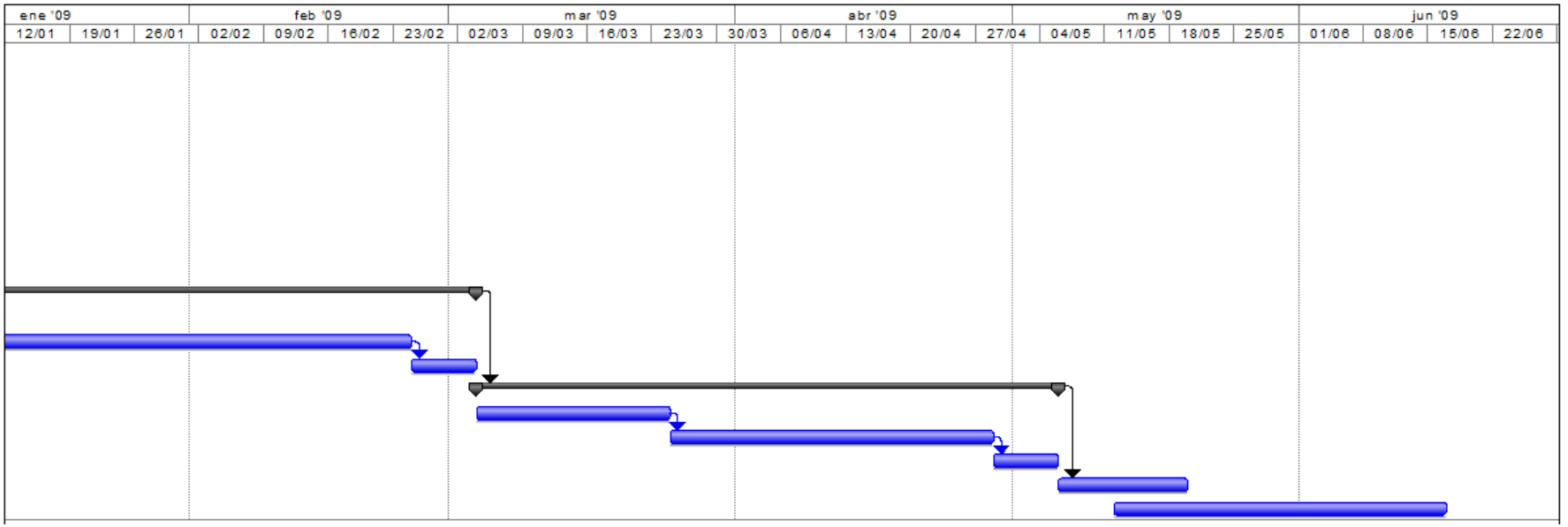

Ilustración 2. Diagrama de Gantt (Parte 2)

2. ESTADO DEL ARTE

2.1 Aplicaciones de Gestión de Administración Pública

2.1.1 Introducción

Según la Comisión Europea de la UE: "La administración electrónica es el uso de las **tecnologías de la información y la comunicación** (TIC) en las administraciones públicas, combinado con cambios organizativos y nuevas aptitudes, con el fin de **mejorar los servicios públicos** y los procesos democráticos y reforzar el apoyo a las políticas públicas" .

La idea clave sobre la administración electrónica es que no se trata simplemente de llevar las TIC a la actividad administrativa, sino que constituye un elemento fundamental en los procesos de **modernización administrativa** de manera que se pueda reducir al mínimo la burocracia.

La ley establece que el **acceso electrónico a la administración** deberá ser posible no más tarde del 31 de diciembre de 2009, pudiendo prorrogar el plazo en caso de administraciones autonómicas o locales cuya disponibilidad presupuestaria no lo permita.

Sin embargo, las administraciones tienen grandes dificultades para disponer de todos sus servicios en la Web. Debido a la flexibilidad de la ley las administraciones, principalmente las autonómicas y locales, pueden obtener prórrogas y por tanto no cumplir con el plazo marcado inicialmente.

2.1.2 Características de la Administración Electrónica

La administración electrónica tiene las siguientes ventajas:

- **Rapidez y comodidad** para los usuarios (a cualquier hora):
  - Evitar colas, desplazamientos y horarios.
  - Acceso a la información más cómodo, fácil y rápido.
  - Acceso universal. La ubicación geográfica y proximidad de oficinas administrativas deja de ser un problema, algo especialmente importante en un país con la población tan dispersa como España.

- Fomento de la participación de los ciudadanos (buzones, cuestionarios, etc.), fomentar una relación interactiva y más positiva.
- **Impulso de la sociedad de la información**: estimula la participación y aprendizaje del ciudadano.
- Simplificación de los procedimientos e integración transparente de diferentes administraciones (soluciones de ventanilla única).
- Para la Administración:
  - **Reducción de costes**, tiempos de tramitación, reducción de errores. Con ello se consigue mayor satisfacción para el funcionariado ya que el peso burocrático del trabajo se puede derivar hacia actividades de asesoramiento y soporte a los ciudadanos y a las empresas. Además, reduce el uso de papel.
  - Mejora de relaciones e **imagen**: transparencia con el ciudadano, entre departamentos y administraciones.
- Impacto en la economía:
  - Según estudios del World Economic Forum "los países que más destacan en cuanto a apertura y eficiencia del sector público y en preparación para la administración electrónica son también los primeros en cuanto a rendimiento económico y competitividad."
  - **Efecto locomotora**: El gasto directo en TIC es considerable, lo que fomenta proyectos con función de piloto para las empresas privadas, así como creación de plataformas y servicios DNIe, aumento de la **confianza** para inversores privados, **aprendizaje**, **inclusión** de ciudadanos y empresas.
  - Se consigue un impulso económico en **regiones geográficamente desfavorecidas**.
  - Menos carga para las empresas, ya que reduce costes y **aumenta su productividad**: contratación electrónica, factura electrónica, modelos de cotización electrónicos en la seguridad social, etc.

Sin embargo, existen problemas para conseguir una implantación de los servicios digitales satisfactoria. Algunos de ellos son los siguientes:

- **Insuficiente penetración de las TIC** en la población española.
- Usabilidad, accesibilidad y falta de experiencia en el uso de las TIC.
- **Desconfianza** en los medios electrónicos de intercambio de información.
- Desconocimiento de la existencia de la Administración online.
- Recelo de la administración con la seguridad electrónica.
- **Falta de integración** entre las diferentes administraciones.
- **Problemas de financiación** y de medios, especialmente para los ayuntamientos pequeños.

2.1.3 Aplicaciones activas

Dentro de las administraciones públicas está aumentando considerablemente el número de trámites disponibles mediante Web. Por ejemplo, las siguientes Webs constan de acceso telemático a los distintos trámites.

- Instituciones Administrativas: Agencia Tributaria, Portal 060, Ministerio de Política Territorial, Oficinas de Atención al Ciudadano (Madrid, Barcelona, Aragón, etc.)
- Instituciones Educativas: Universidad Carlos III de Madrid, Universidad de Málaga, Complejo Universitario de Vigo.
- Instituciones Sanitarias: Instituto de Salud Carlos III
- Otras Instituciones: Dirección General de Tráfico, Fuerzas Armadas Españolas, Patronato Municipal de Deportes, Patronatos Municipales de Turismo, etc.

Estas aplicaciones no son de código abierto, por lo que el acceso a las mismas está restringido y no es posible realizar comparaciones entre ellas.

Sin embargo, sí cabe destacar que la funcionalidad que presentan tiene el denominador común de que **precisan identificar el usuario** que accede a la Web, ya que cada cual tiene acceso únicamente a sus propios datos, y los del resto deben permanecer ocultos.

Además, también es notable que existan diversas administraciones públicas con **necesidades muy parecidas**, como pueden ser las oficinas de atención a los ciudadanos de las distintas provincias. Por ello surgen soluciones que pueden adaptarse a varias situaciones, como la Oficina de Atención al Ciudadano de Intecna, de manera que el código desarrollado pueda ser reutilizado y por tanto se multiplique el beneficio obtenido. Por tanto, en este escenario, es extremadamente útil un código lo más adaptable posible, y que además sea fácil de mantener.

2.2 Registro Telemático

El registro telemático que pretende implementarse es una **evolución del sistema de registro entrada–salida** (E/S) que existe actualmente en todas las administraciones. Por ello se detallará cómo es este servicio, y cómo debe adaptarse a la nueva legislación.

Tradicionalmente, las administraciones reciben solicitudes formales en soporte papel, y éstas son introducidas manualmente por los funcionarios en la aplicación correspondiente. En algunas administraciones también se admiten solicitudes en formato electrónico, pero deben ser introducidas en la aplicación manualmente al igual que las recibidas en papel.

Cada vez que un ciudadano aporte información y documentación se desencadena un proceso integrado por las siguientes acciones:

- El alta en el registro comienza con la aportación de documentación por parte del ciudadano. Este proceso continúa dando entrada en la aplicación entre otros a los siguientes datos:
  - Sentido: Entrada o Salida.
  - Fecha de Presentación.
  - Hora de Presentación.
  - Emplazamiento.
  - Asunto.
  - Clase de Documento.
  - Datos del Remitente:
    - Documento.
    - Tipo persona.
    - Nombre.
    - Primer Apellido.
    - Segundo Apellido.
  - Datos del Destinatario:
    - Tipo destino: Dependencia u Oficina de Registro.
    - Oficina.
    - Observaciones.

Finalizada la introducción de datos, la aplicación de registro genera de forma automatizada el asiento registral de entrada, asignándole un código que permitirá identificar de forma unívoca esta información.

A continuación, se emite un justificante para el ciudadano con los datos de la operación junto con el número de entrada asociado.

- Completado el proceso de alta de una anotación registral de entrada, la aplicación de registro almacenará en la base de datos la información del formulario. En el caso en que el Ciudadano aporte documentación en formato electrónico, el Registro almacenará también dicha documentación.
- En el lugar de destino, una vez recibida la información correspondiente, el funcionario inicia su tramitación. Dicha tramitación puede estar automatizada, en cuyo caso el sistema rellenará los datos generales y el funcionario se encargará de introducir manualmente los datos específicos.

2.2.1 Ejemplos de Registro Telemático

Existen varios ejemplos de soluciones informáticas que cumplen la función de un registro telemático, tanto gratuitas como de pago. Sin embargo, la mayor parte de estas aplicaciones no son autónomas, sino que están integradas en otras mayores que cumplen más funciones.

Un ejemplo de registro telemático autónomo es @RIES, un sistema de información desarrollado por la Junta de Andalucía, basado en la gestión electrónica de documentos. Posee las siguientes características, que pueden generalizarse como funciones básicas de un sistema de este tipo:

- Interconexión informática entre los registros de Entrada/Salida de todos los organismos: Comunicación entre el registro de entrada y el registro o registros del órgano resolutor.
- Implantación de un sistema intercomunicado de registros entre la Administración General del Estado, las Administraciones de las Comunidades Autónomas y las entidades pertenecientes Administración Local.
- Interconexión y transmisión de asientos registrales y, en su caso, de documentos completos entre los distintos Registros.
- Acceso desde cualquier punto remoto a las bases de datos públicas y a otras informaciones de interés para los ciudadanos que proceden de cualquiera de las Administraciones Públicas.
- Posibilidad de realización de trámites administrativos desde el domicilio del ciudadano.

Se podrá emplear ésta u otras aplicaciones similares mediante el conector que se implementará en el servicio de registro telemático del proyecto.

2.3 Firma Electrónica

2.3.1 Introducción

El **artículo 3** de la **Ley 59/2003**, de Firma Electrónica, define la firma electrónica de la siguiente manera:

- 1. La **firma electrónica** es el **conjunto de datos en forma electrónica**, consignados junto a otros o asociados con ellos, que pueden ser utilizados como **medio de identificación** del firmante.
- 2. La **firma electrónica avanzada** es la firma electrónica que permite identificar al firmante **y detectar cualquier cambio ulterior** de los datos firmados, que está vinculada al firmante de manera única y a los datos a que se refiere y que ha sido creada por medios que el firmante puede mantener bajo su exclusivo control.
- 3. Se considera **firma electrónica reconocida** la firma electrónica avanzada basada en un **certificado reconocido** y generada mediante un **dispositivo seguro de creación de firma**.
- 4. La firma electrónica reconocida tendrá respecto de los datos consignados en forma electrónica el **mismo valor que la firma manuscrita** en relación con los consignados en papel.

La firma electrónica nace como una solución para tres problemas distintos que se presentan al tratar con documentos electrónicos. Estos son la confidencialidad, la integridad y la autenticidad:

- **Confidencialidad**: Consiste en permitir el acceso a un documento únicamente a las personas autorizadas.
- **Integridad**: Es el concepto por el cual se garantiza que el documento no ha sido manipulado desde su envío hasta su recepción por el destinatario.
- **Autenticidad**: Consiste en la capacidad de verificar la autoría del documento o el compromiso de cumplir lo expuesto en el mismo.

La autenticidad es el único de los tres problemas que también resuelve la firma autógrafa tradicional. Sin embargo, ésta no garantiza la confidencialidad ni la integridad ya que el documento manuscrito puede ser leído por cualquier persona, y además puede modificarse con posterioridad a la firma.

Dado que incluir un grafo similar a una firma tradicional en un documento electrónico no añadiría ninguna seguridad, se han resuelto los problemas citados mediante la técnica de la criptografía. La criptografía es una rama de las matemáticas que modifica los mensajes de una manera que resuelve los conflictos de autenticidad, confidencialidad e integridad. Básicamente

consiste en un cifrado del mensaje, que permite que sólo pueda ser descifrado por el receptor y asegura que el autor del mensaje puede ser exclusivamente el emisor.

Todos los procesos de firma y cifrado se basan en el uso de **criptografía asimétrica**.

2.3.2 Criptografía asimétrica

Para emplear la criptografía asimétrica es necesario disponer de dos claves, denominadas, **clave privada y clave pública**. En caso de codificar un mensaje con una clave privada, es necesaria la clave pública para descodificarlo, y viceversa.

La clave privada está en posesión de su propietario y debe ser el único que la conoce, mientras que la clave pública se da a conocer abiertamente al resto de los usuarios.

De esta manera se persigue un doble objetivo; por una parte el propietario de la clave puede firmar electrónicamente documentos, ya que cualquier documento que pueda descifrarse con su clave pública tiene que ser necesariamente generado a partir de la clave privada, y por otra el propietario puede recibir documentos cifrados con su clave pública y será el único que podrá descifrarlos.

2.3.3 Certificados digitales

Un certificado digital es un **documento digital que identifica a una persona** con sus correspondientes claves criptográficas. Este tipo de certificado contiene la siguiente información:

- Identificación del titular.
- Distintivos del certificado: número de serie, entidad que lo emitió, fecha de emisión, periodo de validez, etc.
- Una pareja de claves: pública y privada.
- La firma electrónica del certificado con la clave de la autoridad de certificación que lo emitió.

Toda la información es pública, excepto la clave privada, que nunca es cedida por su propietario, y que debe mantener en secreto para no comprometer la seguridad del cifrado.

2.3.4 Firma electrónica

Una firma electrónica es un resumen que garantiza la integridad del mensaje original. Para ello se extrae a partir del mensaje una huella digital mediante un algoritmo que garantiza que es imposible volver a obtener el mensaje original a partir de dicha huella, y además, si cambia el mensaje, la huella obtenida es diferente.

2.3.4.1 ¿Cómo se genera una firma electrónica?

- Se obtiene una huella digital del documento digital que se quiere firmar.
- Se cifra la huella obtenida mediante criptografía asimétrica la clave privada del certificado.
- Se crea un documento que incluye la huella digital cifrada y la parte pública del certificado y, opcionalmente, el documento original.
- Se firma el nuevo documento obtenido.

2.3.4.2 ¿Cómo se verifica una firma electrónica?

- Se descifra la huella digital cifrada con la clave privada mediante la clave pública del certificado.
- Se obtiene la huella digital del documento original mediante el mismo algoritmo que empleó el remitente.
- Se comprueba si las huellas digitales coinciden.
- Se consulta a la autoridad de certificación emisora la validez del certificado para garantizar la autenticidad del origen de la firma.

2.3.5 Ejemplos de Firma Electrónica

Existen diferentes ofertas de servicios de firma electrónica en la Web, tanto privados como públicos.

Por ejemplo, @FIRMA es la plataforma corporativa de la Junta de Andalucía para autenticación y firma electrónica. Mediante esta herramienta se pueden realizar procesos de autenticación y firmado digital a través del uso de certificados digitales, independientemente del entorno de desarrollo en que hayan sido programados.

Es una herramienta de uso libre, y actualmente está encargado de su mantenimiento el Ministerio de Administraciones Públicas conjuntamente con la Junta de Andalucía.

Por otra parte existen soluciones privadas como la que aporta TB-Solutions con su plataforma de firma electrónica ASF. Esta herramienta también posee la capacidad de firmar en diferentes formatos, así como de comprobar la validez de certificados digitales. Asimismo permite constituir una autoridad de certificación.

Al contrario de lo que ocurre con los registros telemáticos, los servicios de firma electrónica sí se ofrecen de manera autónoma, principalmente porque el uso de este tipo de servicios se extiende a muchos más ámbitos.

2.4 Java

La tecnología Java está formada por dos componentes: El lenguaje de programación y la plataforma.

El lenguaje Java es un **lenguaje de programación orientado a objetos**, desarrollado por Sun MicroSystems, en 1991. Su sintaxis guarda mucha semejanza con C y C++, pero tiene un modelo de objetos más simple y elimina herramientas de bajo nivel, que suelen inducir a muchos errores, como el manejo de punteros o de memoria.

La plataforma Java es capaz de ejecutar aplicaciones desarrolladas usando el lenguaje de programación Java, que se compilan creando un código intermedio independiente de la plataforma denominado bytecode.

Java está diseñado para que un programa escrito en este lenguaje sea ejecutado **independientemente de la plataforma**, tanto hardware, como software o sistema operativo. Esta portabilidad se consigue haciendo de Java un lenguaje en parte interpretado y en parte compilado.

En casi todos los casos las plataformas son descritas como la combinación del sistema operativo y el hardware. La plataforma Java se diferencia de estas plataformas en que es una plataforma sólo de software, y se ejecuta sobre las otras plataformas de hardware. La plataforma Java tiene dos componentes: La máquina virtual de Java o Java Virtual Machine (JVM) y el Java API (Application Programming Interface).

JVM es dependiente de la plataforma y es capaz de interpretar y ejecutar instrucciones expresadas en Java bytecode, que es generado por el compilador del lenguaje Java. La gran ventaja de la máquina virtual java es que aporta portabilidad al lenguaje de manera que el bytecode escrito en un entorno, por ejemplo Linux, puede ser interpretado en otro, como Windows. Tan sólo es necesario disponer de la máquina virtual para dichos entornos.

Java API es una gran **colección de componentes de software** que proporcionan muchas utilidades para el programador, por ejemplo; funcionalidad para implementar interfaces gráficas, manejadores de bases de datos, etc.

2.5 Librerías, extensiones y frameworks

Java es una plataforma de código abierto que se encuentra entre las más utilizadas actualmente, lo que ha propiciado el nacimiento de una comunidad de desarrolladores muy activa, que ha creado diversas **herramientas de uso público** que resuelven los problemas más recurrentes al enfrentarse con aplicaciones software.

Según la presentación de estas herramientas se denominan de la siguiente manera:

- **Librería**: El término librería o “toolkit” se refiere a una herramienta encapsulada en uno o varios archivos independientes del programa que hace uso de ellos.

- **Extensión**: En el dominio del desarrollo software, una extensión consiste en una aplicación que incorpora nueva funcionalidad a una herramienta ya existente.
- **Framework**: Es un conjunto de herramientas, que permite organizar y desarrollar un proyecto software.

A continuación se detallan las diversas utilidades que se han empleado en el desarrollo del proyecto.

2.5.1 Java Database Connectivity

Para acceder a la información almacenada en una base de datos es necesaria la utilización de un **lenguaje específico para cada base de datos** distinto a Java. Existen diversos frameworks y librerías que integran Java con los lenguajes de acceso a base de datos y ayudan al programador a acceder a los datos de una de una forma más cómoda, como por ejemplo Java DataBase Connectivity (JDBC) y Java Persistence API (JPA).

En este caso se empleará la tecnología JDBC**,** que es un API que permite la **ejecución de operaciones** sobre bases de datos desde el lenguaje de programación Java de manera **transparente** al sistema operativo donde se ejecute y a la base de datos accedida. Para ello, JDBC utiliza el dialecto SQL propio de la base de datos.

El API JDBC se presenta como una colección de interfaces Java y métodos de gestión de manejadores de conexión de cada modelo específico de base de datos. Estos manejadores consisten en implementaciones de las interfaces Java que adquieren el localizador o Uniform Resource Locator (URL) de la base de datos a la que van a acceder.

Para utilizar una base de datos se ejecuta la aplicación, que contendrá la librería de conexión apropiada. Durante la ejecución se provee el localizador a la base de datos y los parámetros de conexión específicos para establecer una conexión. Una vez obtenida, se pueden realizar en la base de datos las tareas para las que se disponga permiso: consultas, actualizaciones, incluyendo creación, modificación y borrado de tablas, ejecución de procedimientos almacenados en la base de datos, etc.

2.5.2 Hibernate

Hibernate es una herramienta creada para la plataforma Java, aunque ya está extendida a otras como .NET, que permite la **correspondencia entre el modelo orientado a objetos y el modelo relacional**. Con ella, se facilita el mapeo de atributos entre una base de datos relacional tradicional y el modelo de objetos de una aplicación, mediante archivos declarativos (XML) que permiten establecer estas relaciones. Está distribuido bajo los términos de la licencia GNU LGPL.

Hibernate se emplea cuando en una misma aplicación coexisten dos modelos diferentes de datos: el propio de la aplicación, el modelo orientado a objetos; y el que emplea la base de datos, el modelo relacional. Para ello el desarrollador debe especificar cómo es el modelo de

datos, incluyendo los objetos que lo componen, sus atributos, las relaciones entre ellos y sus cardinalidades, etc.

Una vez detalladas todas estas cuestiones, la aplicación puede **interactuar con la base de datos como si ésta emplease un modelo orientado a objetos**, de manera transparente al programador. Esto lo consigue generando automáticamente las sentencias SQL y manejando las respuestas de la base de datos, que se convierten a objetos.

Hibernate permite definir en primer lugar el modelo orientado a objetos y generar automáticamente el modelo relacional y viceversa. También ofrece un lenguaje de consulta de datos denominado Hibernate Query Language (HQL) y un API denominado "Criteria" que permite crear consultas directamente en el código Java.

Como contrapartida, Hibernate aumenta el tiempo de ejecución de las consultas a la base de datos ya que tiene menor flexibilidad que la ejecución directa de sentencias SQL, por lo que no es recomendable en sistemas cuyo tiempo de respuesta sea crítico.

2.5.3 Castor

El Framework Castor es una herramienta de código abierto que **transforma datos** desde **XML a objetos Java** y viceversa, así como la integración de los datos con las bases de datos.

Castor está compuesto de los siguientes módulos, independientes entre sí:

- **Castor XML**: Este módulo se emplea para transformar objetos Java a archivos XML y viceversa. A este proceso se le denomina mapeado.

  Para ello puede emplear convenciones de mapeado para clases simples, o archivos en los que se defina el mapeo para clases más complejas.

- **Castor XML Code Generator**: Este módulo sirve para generar las clases Java a partir de un esquema XML

- **Castor JDO Persistence framework**: éherramienta se utiliza para transformar **objetos Java** a instancias de una **base de datos relacional**.

Además, incluye herramientas que permiten crear una clase de mapeo a partir de clases java existentes, o transformar esquemas DTD a XML Schema. Asimismo, Castor también incluye funcionalidad que facilita la integración con otros frameworks.

2.5.4 Junit

JUnit es un conjunto de bibliotecas creadas para hacer **pruebas unitarias** en aplicaciones Java, aunque ya están disponibles para otras tecnologías como .NET.

Este framework permite realizar la ejecución de las clases Java de manera controlada, permitiendo al desarrollador comprobar si el comportamiento de cada método es correcto. Para ello se realizan pruebas que consisten en determinar si, dado un valor de entrada para cada método, el valor de retorno es el esperado. En caso de resultar correcto, JUnit indica que la prueba es correcta, y en caso contrario especifica qué tipo de error se ha producido.

Existen tres maneras de presentar los resultados o "runners": modo texto, modo gráfico o como tarea en Ant. Estas presentaciones pueden integrarse en herramientas de desarrollo como NetBeans o Eclipse, que además permiten realizar automáticamente las pruebas, y que incluso cuentan con plug-ins que generan automáticamente las plantillas sobre las que desarrollar las pruebas.

2.5.5 Xfire

Codehaus XFire es un framework de Java diseñado para **desarrollar y consumir servicios Web**.

Proporciona un **alto rendimiento** porque está construido en un modelo de baja memoria basado en Streaming API for XML (StAX). Además, está construido con un API sencillo que puede ser embebido.

XFire provee, entre otras, las siguientes funcionalidades:

- Todos los estándares de servicios web incluyendo SOAP, WSDL, etc.
- Distintas capas de transporte: HTTP, JMS, XMPP, In-JVM, etc.
- Soporte para Spring y Castor.

2.5.6 SpringFramework

Spring Framework o Spring es una herramienta de código abierto para Java, aunque también existe para .Net, que facilita el desarrollo de funcionalidades específicas para la **manipulación de objetos**, tanto cuando se emplean Enterprise JavaBeans (EJB) como en caso contrario.

Ofrece mucha libertad para el desarrollo, además de emplear soluciones para prácticas comunes en la industria fáciles de llevar a la práctica y bien documentadas.

Estas características han conseguido que Spring sea popular en la comunidad de programadores de Java como sustituto del model,o de EJB.

Su popularidad ha fomentado que se creen muchas extensiones que amplían la funcionalidad, permitiendo, por ejemplo, construir aplicaciones basadas en servicios web.

2.6 Servicios web

2.6.1 ¿Qué es un servicio web y porque se debe usar?

Se conoce como **servicio Web** a una colección de **protocolos** y estándares que sirven para **intercambiar datos** entre las aplicaciones.

Para llevar a cabo esta tarea se emplean estándares abiertos. Las organizaciones OASIS y W3C son las encargadas de la arquitectura y la reglamentación de los servicios Web. Además, se ha creado el organismo WS-I para desarrollar diversos perfiles para definir de manera más exhaustiva estos estándares.

Un servicio Web está **definido** por un **URI** (Uniform Resource Identification) y por su **interfaz**, a través de la cual se puede acceder a él.

Para emplear un servicio Web, ya sea por parte de un sitio Web o una aplicación, se realiza una **petición en SOAP** (Simple Object Access Protocol). SOAP es un protocolo que define los mensajes mediante los cuales se realizará la comunicación. Estos mensajes están escritos en XML, y se intercambian bidireccionalmente entre servidor y cliente.

Por otra parte, para **definir** qué **mensajes** se utilizan para consumir un determinado servicio, se emplea el formato **WSDL** (Web Services Description Language). Al igual que en SOAP, los mensajes también están escritos en XML.

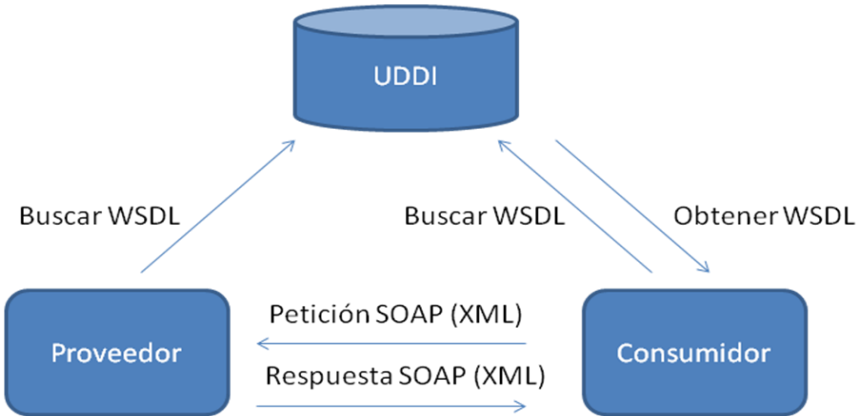

Ilustración 3. Estructura de un servicio web

2.6.2 Ventajas de los servicios web

Los servicios Web aportan, por la manera en la que están definidos, varias ventajas frente a otros sistemas de comunicación.

En primer lugar, la comunicación entre aplicaciones es **independiente del lenguaje de programación** y de la plataforma en la que se ejecutan los programas. Esto implica mayor facilidad para intercambiar datos en redes de ordenadores, incluyendo tanto redes internas de una empresa como Internet de forma global.

Por otra parte, los protocolos que se siguen son estándares y abiertos, con lo que se garantiza **interoperabilidad** plena entre distintas aplicaciones.

Por último, los servicios Web permiten construir una **aplicación de manera modular**, en lo que se conoce como **arquitectura orientada a servicios** (SOA). De este modo, se desarrollan pequeños módulos que cumplen un servicio, de manera que pueden ser empleados por otros servicios. Así, se consiguen aplicaciones más flexibles y reutilizables.

2.6.3 Inconvenientes de los servicios web

Los inconvenientes de los servicios Web vienen dados, principalmente, por la **poca madurez** de los mismos frente a otros sistemas de comunicación como puede ser CORBA. De esta circunstancia puede derivar que surjan nuevos problemas no resueltos con el estándar, o que los desarrolladores no se encuentren familiarizados con él.

Por otra parte, puede ofrecer **menos rendimiento** ya que los mensajes están basados en texto.

2.7 Estándares

2.7.1 XML

La creación de **XML** (eXtended Markup Language) surge de la necesidad de dotar a la Web de una **estructura más semántica**. Mediante este lenguaje, cada programador puede disponer de sus propias etiquetas para representar semánticamente un dominio.

Además, permite estructurar las páginas Web como **árboles de etiquetas**, ya que unas pueden estar contenidas en otras. Mediante este mecanismo se aumenta la capacidad de representación semántica del lenguaje.

A sus funcionalidades básicas hay que sumarle el extenso número de complementos que permiten el desarrollo ágil de XML. Entre los más destacados están los siguientes:

- **Herramientas para la definición de la estructura del documento**: Mediante estas herramientas pueden especificarse qué etiquetas pueden utilizarse y cuáles están prohibidas. Además, también se puede especificar la estructura que tienen que seguir las etiquetas. Entre estas herramientas destacan DTD (Document Type Definition) y XML Schema.

- **Herramientas para el diseño del documento**: Estas herramientas permiten, de una manera sencilla, modificar la representación visual de un documento XML. Las más utilizadas son CSS (Cascade Style Sheet) y XSL (eXtended Style Language)
- **Herramientas para el tratamiento de documentos**: Estas herramientas permiten editar los documentos XML, añadiendo, modificando o eliminando etiquetas. Algunas de estas herramientas son SAX (Simple API for XML) y DOM (Document Object Model)

2.7.2 SOAP

SOAP es un **protocolo de comunicaciones** creado por Microsoft, IBM y otros basándose en el protocolo XML-RCP, y actualmente está mantenido por W3C. Emplea mensajes XML adaptados a una estructura definida con los que se realizan peticiones y respuestas, o códigos de error si procede.

Los servicios Web emplean el protocolo SOAP para proporcionar acceso a los objetos remotos, asegurando así la independencia entre la comunicación, la plataforma de la aplicación y el lenguaje de programación empleado. De la misma manera permite el uso de **varios protocolos de transporte**, aunque principalmente se emplea con HTTP.

Una gran ventaja de SOAP es la gran aceptación que tiene en el mundo empresarial, ya que es uno de los protocolos de interconexión más empleados, y, a pesar de no ser actualmente un estándar, es tratado como tal.

Desde el 27 de Abril de 2007 está disponible SOAP versión 1.2.

La estructura de un mensaje de este tipo se divide en tres partes:

- **Envelope**: Elemento raíz que define el contenido del mensaje.
- **Header** (Opcional): Contiene información sobre el mensaje. Se puede emplear para asignar ciertas características al mensaje.
- **Body**: En esta sección se incluyen las peticiones, las respuestas y los códigos de error. Estos últimos deben incluirse en un campo Fault.

2.7.3 WSDL

WSDL es un lenguaje basado en XML en el que se determinan aspectos sobre la **interfaz**, la **semántica** y la **administración** de un **servicio Web.**

Principalmente, en un fichero WSDL se incluirá la **dirección del servicio** así como las **llamadas a las funciones**. Esto permitirá acceder al mismo tanto a otras aplicaciones como a otros servicios.

WSDL es extensible y se puede utilizar para describir múltiples servicios, incluyendo SOAP pero también DCOM sobre UDP. Dado que la tendencia actual es describir las comunicaciones de forma ordenada, WSDL puede ser la solución para nuevos modelos de comunicación que puedan surgir.

2.7.4 POJO

Se denomina Plain Old Java Object (POJO) para referirse a aquellas **clases Java simples que no dependen de ningún framework**.

POJO no es una tecnología en sí misma, sino que se emplea para enfatizar la programación orientada a objetos, más pura, en contraposición de otra técnica como son los estándares EJB anteriores al 3.0, en los que los "Enterprise JavaBeans" deben implementar interfaces especiales.

Este tipo de programación surgió como respuesta de Java al aumento de la cuota de mercado de otros lenguajes orientados a objetos, como Ruby y Python.

3. MODELO

3.1 Introducción

El objetivo de esta etapa del proyecto es obtener una especificación detallada del sistema propuesto, que sirva de base para los diseñadores.

Esta fase del proyecto, consiste en **definir el problema** entendiendo el dominio de aplicación y su funcionalidad. El objetivo es que dicho software responda a las necesidades de aquel al que va destinado, para resolverlo en el diseño del sistema con posterioridad.

3.2 Descripción General

La aprobación de la Ley 11/2007, de 22 de junio, de Acceso Electrónico de los ciudadanos a los Servicios Públicos, ha supuesto la incorporación de España al reducido grupo de países que disponen de leyes para proteger los derechos digitales de sus ciudadanos: Estados Unidos, Finlandia, Francia, Austria e Italia.

Este precepto obliga a las Administraciones a emplear una serie de aplicaciones que automaticen todos los procedimientos administrativos de la organización.

Las aplicaciones deben controlar los tiempos y plazos, la identificación de los órganos responsables de los procedimientos, la tramitación ordenada de los expedientes electrónicos y facilitar la simplificación y la publicidad de los procedimientos. Además deberán incorporar e integrarse con los servicios básicos de Administración Electrónica: firma electrónica, registro telemático, notificaciones telemáticas, archivo electrónico, intercambio de certificados, pasarela de pagos, etc.

El sistema que se desea construir satisfará las necesidades de la administración pública de ofrecer los **servicios de registro telemático y firma electrónica**. Para conseguirlo, se implementarán dos servicios completamente nuevos, sin reutilizar componentes anteriores.

Estas aplicaciones están **orientadas a formar parte de diferentes sistemas mayores**, por lo que deben ser **fácilmente reutilizables**.

Por otra parte empleará servicios proveídos por otros módulos, por lo que tiene que ser **capaz de comunicarse**. Además, estos módulos pueden cambiar, y estos cambios deberán ser lo más transparentes posibles a la aplicación.

3.3 Servicio Web de Registro Telemático

3.3.1 Introducción

El registro telemático consiste en una **puerta telemática formal** que concentre e identifique todas las entradas que se producen por este canal entre la Administración y los ciudadanos. Este canal telemático identifica todas las comunicaciones con un número único.

Tendrá la misma **validez legal** que el servicio prestado por el registro presencial de entrada y salida, realizando las mismas funciones que éste, pero haciéndolo mediante el canal telemático.

3.3.2 Objetivo

El objetivo de este servicio es permitir a los ciudadanos la presentación por vía telemática de cualquier tipo de documentación requerida en relación a la Administración.

El registro telemático se hará cargo de la recepción y validación de las solicitudes, así como de la entrega de justificantes de registro.

Las principales funciones del registro telemático son:

- La **recepción y remisión de solicitudes**, escritos y comunicaciones relativas a los trámites y procedimientos que se especifiquen en su norma de creación y que sean competencia del órgano que creó el Registro.
- La anotación de los correspondientes **asientos de entrada y salida**.
- El registro telemático emitirá un mensaje de **confirmación de la recepción de la solicitud**, escrito o comunicación, en el que constarán los datos proporcionados por el interesado, junto con la acreditación de la fecha y hora en que se produjo la recepción y una clave de identificación de la transmisión. Dicho mensaje tendrá el valor de **recibo de presentación**.

El registro telemático estará **operativo las 24 horas del día, todos los días del año**.

Se empleará en dos casos básicos: Entrada y Salida, por lo que se encargará tanto de gestionar los formularios recibidos como de interaccionar con los distintos servicios y sistemas.

3.3.3 Contexto

El módulo de registro telemático estará integrado dentro del un si**stema orientado a servicios** y proporcionará un **servicio horizontal a los restantes módulos**. Además empleará los servicios de seguridad, firma electrónica y custodia documental.

3.3.4 Diagrama de casos de uso

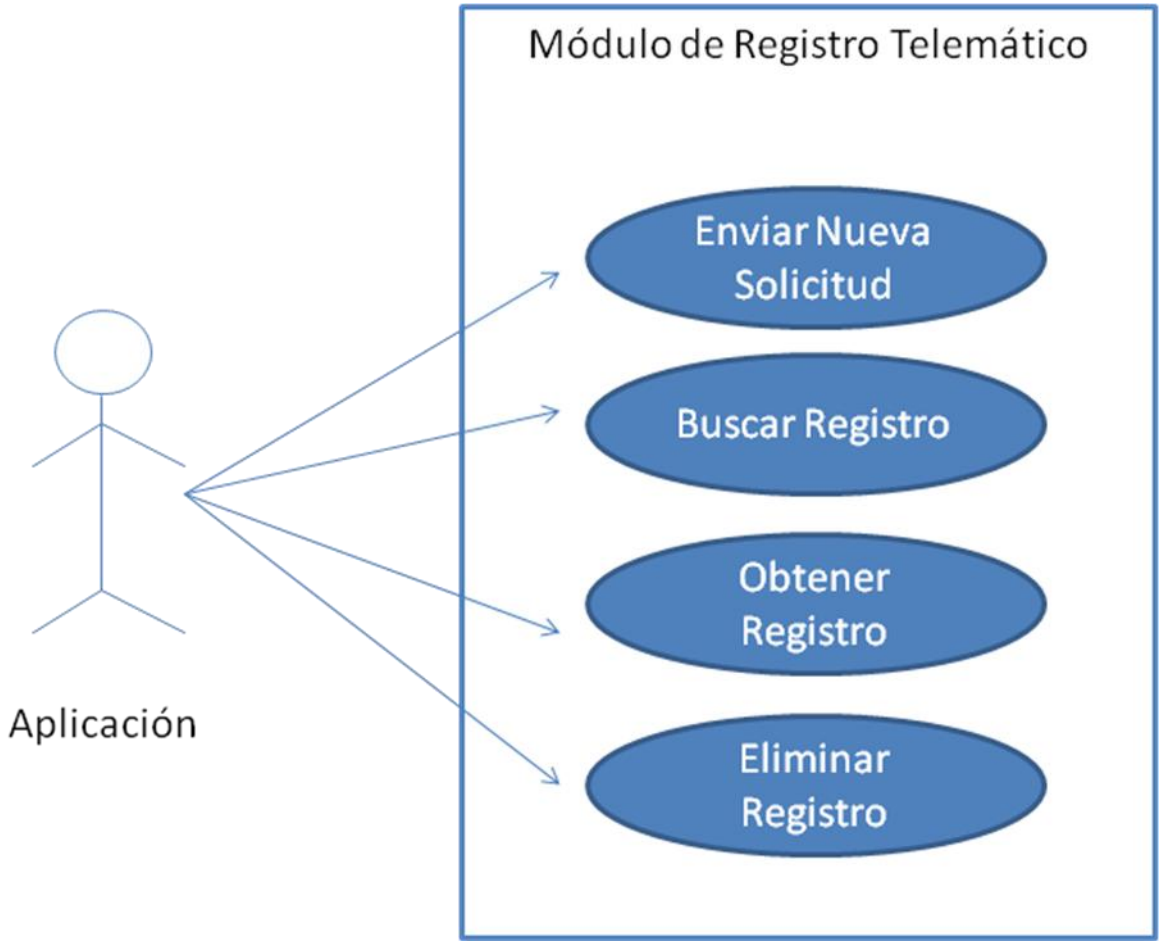

Ilustración 4. Caso de Uso: Módulo de Registro Telemático

3.3.5 Casos de uso

3.3.5.1 Caso de uso Envío Nueva Solicitud

La funcionalidad de enviar una nueva solicitud consistirá en la acción de recibir un envío de una solicitud por parte de la aplicación usuaria, procesar dicho envío y devolver un justificante de la petición.

Durante el proceso, se comprobará que la firma digital de la solicitud es correcta. Este mismo documento se firmará digitalmente para generar el justificante.

En caso de existir documentos anexos, éstos se almacenan mediante una petición al servicio de custodia documental.

Propósito	La aplicación usuaria envía una Nueva Solicitud al Registro Telemático
Precondiciones	Precondiciones: • Disponibilidad plataforma de registro telemático Parámetros de entrada:

	• Identificador de la aplicación usuaria • Tipo de solicitud: entrada o salida • Solicitud de Registro XML firmada • Documentos Anexos • Hash de los Anexos • Ticket que identifica el emisor con la aplicación .
Postcondiciones	Resultado del proceso: • Resultado del envío de nueva solicitud de registro: ▪ Envío Solicitud Correcto ▪ Envío Solicitud Incorrecto ▪ Error • Descripción del resultado: ▪ En caso de error: descripción del error
Circunstancias de uso	• Uso online • Servicio restringido

Tabla 1. Caso de Uso: Envío de Nueva Solicitud

3.3.5.1.1 Diagrama de secuencia

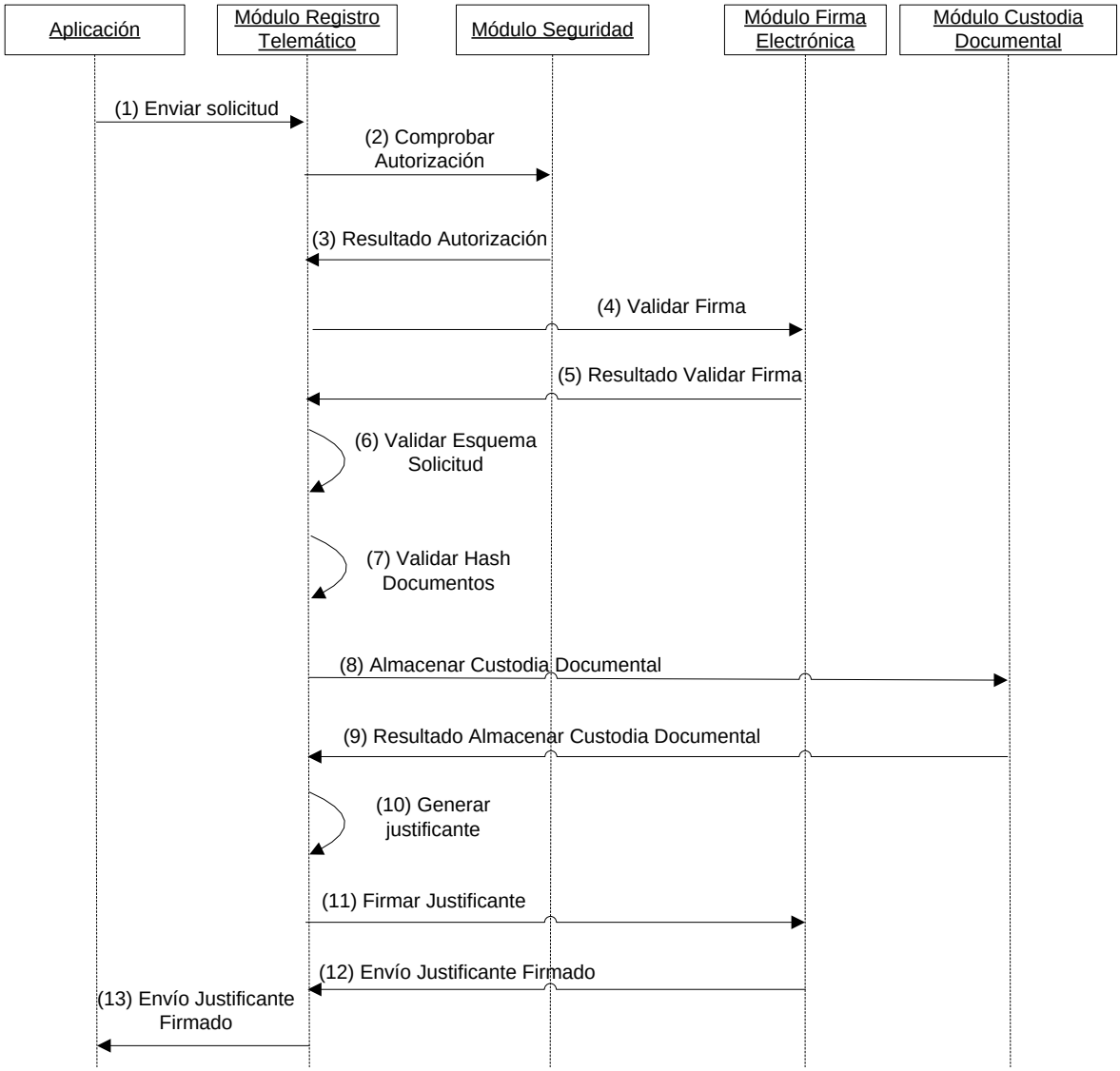

Ilustración 5. Diagrama de Secuencia: Módulo de Registro Telemático

- **Enviar solicitud**: La Aplicación Usuaria envía la solicitud de registro (XML) firmada junto con los documentos anexos.
- **Comprobar autorización**: El Módulo Registro Telemático solicita la validación del ticket al Módulo Seguridad.
- **Resultado de la autorización**: Se devuelve un mensaje informando del resultado.
- **Validar firma**: El Registro Telemático envía la solicitud al Módulo de Firma para que realice la validación de la firma y del certificado firmante.

- **Resultado validar firma**: El Módulo de Firma envía el resultado de la comprobación al Registro Telemático.
- **Validar esquema solicitud**: El Registro Telemático valida que el esquema de la solicitud sea correcto.
- **Validar hash documentos**: El Registro Telemático valida que los hash de los documentos anexos sean correctos.
- **Almacenar en Custodia Documental**: El Registro Telemático envía la solicitud de registro a Custodia Documental para almacenarla junto con sus anexos.
- **Resultado Almacenar en Custodia Documental**: El módulo de Custodia Documental devuelve el resultado.
- **Generación justificante**: Una vez terminado el registro de la comunicación enviada se devolverá un código de finalización, el número de entrada al Registro Telemático, la fecha y la hora de recepción, la fecha y la hora de entrada y un justificante firmado que contenga todos los datos del registro. Este justificante incluirá un sello del tiempo.
- **Firmar Justificante**: El Registro Telemático envía el justificante al Módulo de Firma para firmarlo en el servidor.
- **Envío Justificante Firmado**: El Módulo de Firma devuelve al Registro Telemático el justificante firmado.
- **Envío Justificante Registro**: El Registro Telemático envía el justificante de registro firmado a la Aplicación Usuaria.

3.3.5.2 Caso de uso Búsqueda Registros

El caso de uso de Búsqueda de Registros consistirá en la petición por parte de la aplicación usuaria de información sobre registros que cumplan los criterios de búsqueda.

El usuario dispone de varios campos para delimitar la búsqueda. Los campos vacíos indicarán que cualquier registro coincide con la búsqueda.

La aplicación de registro telemático permitirá el acceso a la información por parte de las aplicaciones autorizadas, pero no controlará los permisos de acceso, por lo que deberá ser la capa superior la que limite las búsquedas según el rol del usuario.

El resultado será el conjunto de los números de identificador de registro que coinciden con los parámetros.

Propósito	La aplicación usuaria puede consultar las solicitudes registradas, filtrando por los criterios de búsqueda definidos
Precondiciones	Precondiciones: • Disponibilidad plataforma de registro telemático Parámetros de entrada: • Identificador de la aplicación usuaria • Criterios de búsqueda del registro ▪ Número de Registro ▪ Número de Expediente ▪ Identificador de Aplicación de consulta ▪ NIF Remitente ▪ Nombre Remitente ▪ Fecha Recepción Desde ▪ Fecha Recepción Hasta ▪ Código de Destino ▪ Código de Asunto • Ticket que identifica el emisor con la aplicación
Postcondiciones	• Resultado de la consulta: XML con listado de registros que cumplen los criterios de búsqueda. El XML contendrá únicamente los números de registro • Descripción del resultado: ▪ En caso de error: descripción del error
Circunstancias de uso	• Uso online • Servicio restringido

Tabla 2. Caso de Uso: Búsqueda de Registros

3.3.5.2.1 Diagrama de secuencia

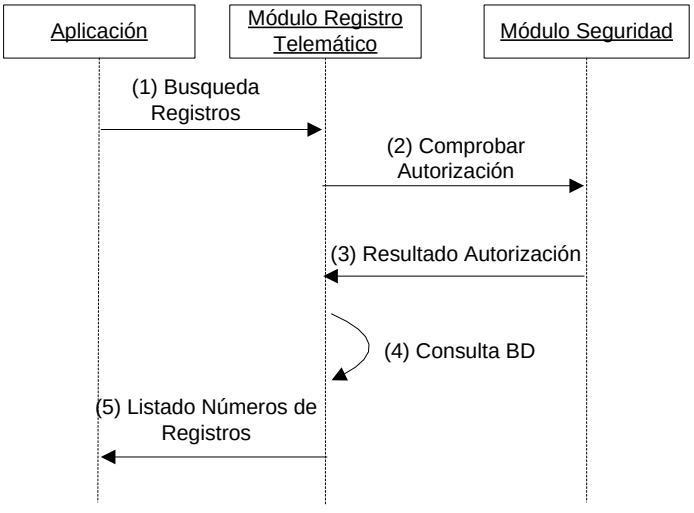

Ilustración 6. Diagrama de Secuencia: Buscar Registros

- **Búsqueda Registros:** La Aplicación Usuaria solicita la búsqueda los registros que cumplan ciertas condiciones.
- **Comprobar autorización**: El Módulo Registro Telemático solicita la validación del ticket al Módulo Seguridad.
- **Resultado de la autorización**: Se devuelve un mensaje informando del resultado.
- **Consulta BBDD**: El Módulo Registro telemático consulta en la base de datos los registros correspondientes.
- **Listado Números de Registros**: Se devuelven los registros que coinciden con el filtro de búsqueda.

3.3.5.3 Caso de uso Obtener Registro

El caso de uso de Obtener Registro consiste en la funcionalidad de obtener un único registro a partir de su identificador. A diferencia del caso de uso de obtener registros, se obtienen todos los datos del registro

Además, puede obtener también los documentos anexos asociados a la solicitud de registro.

Al igual que el caso de uso de obtener registros, no se encarga de comprobar los permisos de acceso del usuario, por lo que es el recubrimiento el que debe garantizar la confidencialidad.

Propósito	La aplicación usuaria puede obtener información detallada de un registro
Precondiciones	Precondiciones: • Disponibilidad plataforma de registro telemático Parámetros de entrada: • Identificador de la aplicación usuaria • Número de registro • Devuelve Documentos (Booleano: 0 si no se desea obtener los documentos anexos a la solicitud de registro, y 1 en caso contrario) • Ticket que identifica el emisor con la aplicación
Postcondiciones	• Resultado de la consulta: XML con los datos de Registro: ▪ Número de Registro ▪ Fecha de Recepción ▪ Nombre Remitente ▪ Código del Destino ▪ Código del Asunto ▪ Código de Aplicación ▪ Número de Expediente ▪ Datos de los documentos anexos, en caso de ser solicitado • Descripción del resultado: ▪ En caso de error: descripción del error
Circunstancias de uso	• Uso online • Servicio restringido

Tabla 3. Caso de Uso: Obtener Registro

3.3.5.3.1 Diagrama de secuencia

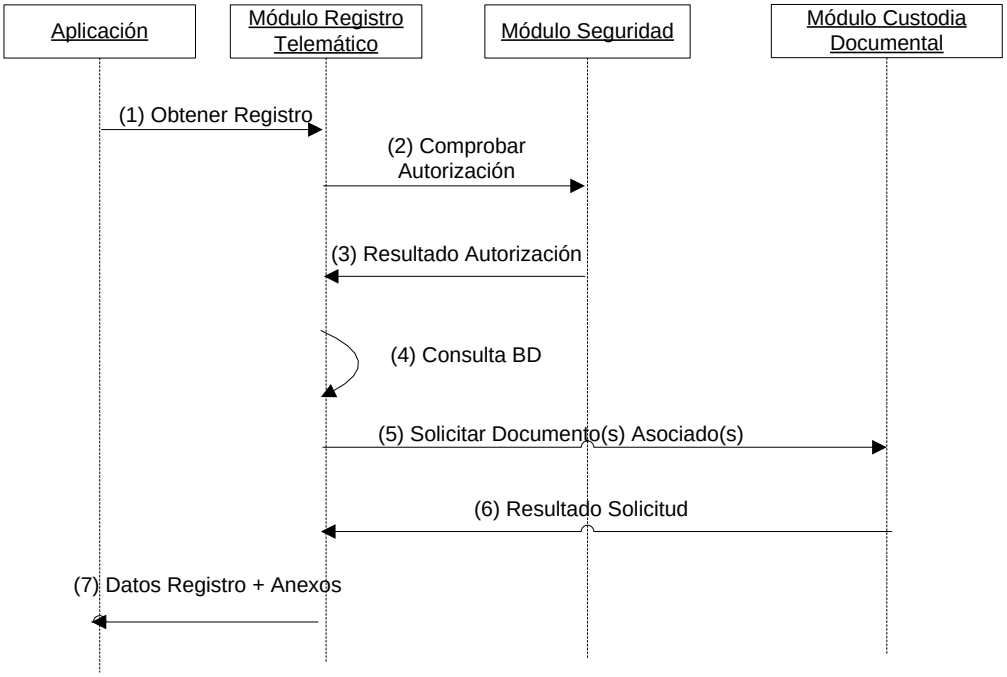

Ilustración 7. Diagrama de Secuencia: Obtener Registros

- **Obtener Registro:** La Aplicación Usuaria solicita la búsqueda de un registro
- **Comprobar autorización**: El Módulo Registro Telemático solicita la validación del ticket al Módulo Seguridad.
- **Resultado de la autorización**: Se devuelve un mensaje informando del resultado.
- **Consulta BBDD**: El Módulo Registro Telemático consulta en la base de datos el registro correspondiente.
- **Solicitar documento(s) asociado(s)**: El Módulo Registro Telemático solicita al Módulo de Custodia Documental los documentos anexos asociados
- **Resultado solicitud**: El Módulo de Custodia Documental devuelve los documentos solicitados
- **Resultado**: Se devuelve el detalle del registro.

3.3.5.4 Caso de uso Eliminar Registro

Las aplicaciones usuarias pueden eliminar registros del Registro Telemático, indicando el número de registro, identificador de la aplicación y el tipo de registro (entrada o salida).

El borrado es de tipo lógico, es decir, se marca el registro como borrado pero no se elimina de la base de datos.

Propósito	Eliminar registro
Precondiciones	Precondiciones: • Disponibilidad plataforma de registro telemático Parámetros de entrada: • Identificador de la aplicación usuaria • Identificador del registro a eliminar • Ticket que identifica el emisor con la aplicación
Postcondiciones	Resultado del proceso: • Resultado de la validación: ▪ Registro eliminado ▪ Error • Descripción del resultado: ▪ En caso de error: descripción del error
Circunstancias de uso	• Uso online • Servicio restringido

Tabla 4. Caso de Uso: Eliminar Registro

3.3.5.4.1 Diagrama de secuencia

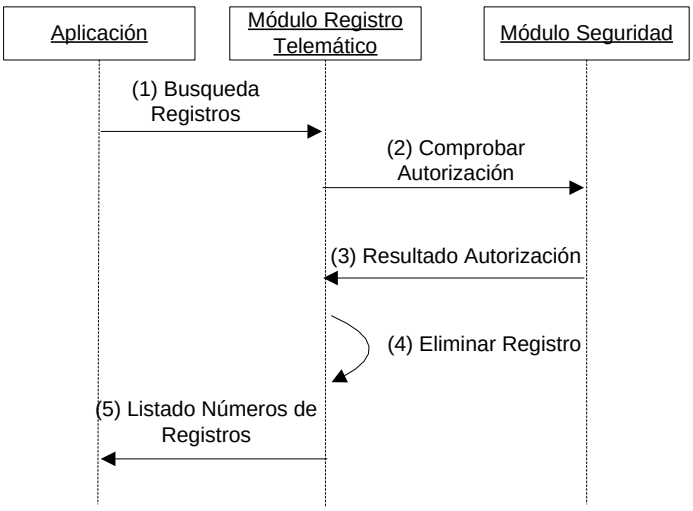

Ilustración 8. Diagrama de Secuencia: Eliminar Registro

- **Búsqueda Registros**: La Aplicación Usuaria solicita la búsqueda de un registro para eliminarlo

- **Comprobar autorización**: El Módulo Registro Telemático solicita la validación del ticket al Módulo Seguridad.
- **Resultado de la autorización**: Se devuelve un mensaje informando del resultado.
- **Eliminar Registro**: La Aplicación Usuaria solicita al Registro Telemático la eliminación de un registro.
- **Resultado**: Se devuelve un mensaje informando del resultado

3.3.6 Interfaces del sistema

3.3.6.1 Interfaces con el Módulo de Firma

3.3.6.1.1 Validar firma

El módulo de Registro Telemático se comunicará, en caso necesario, con el módulo de firma electrónica para validar la firma.

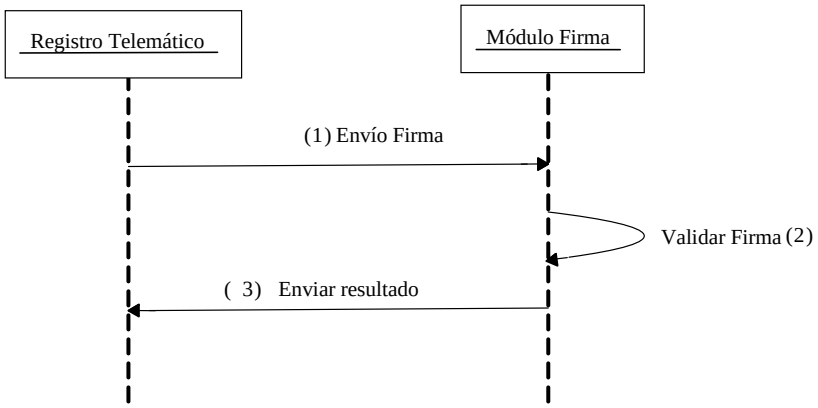

Ilustración 9. Diagrama de Secuencia: Validar Firma

- **Envío Firma**: El módulo Registro Telemático envía la firma al módulo de firma, para que éste lo valide.
- **Validar Firma:** El módulo de firma comprueba la validez de la firma
- **Enviar Resultado**: El módulo de firma devuelve la respuesta

3.3.6.1.2 Firmar servidor

El módulo de Registro Telemático se comunicará, en caso necesario, con el módulo de firma electrónica para firmar digitalmente los justificantes generados al recibir una solicitud.

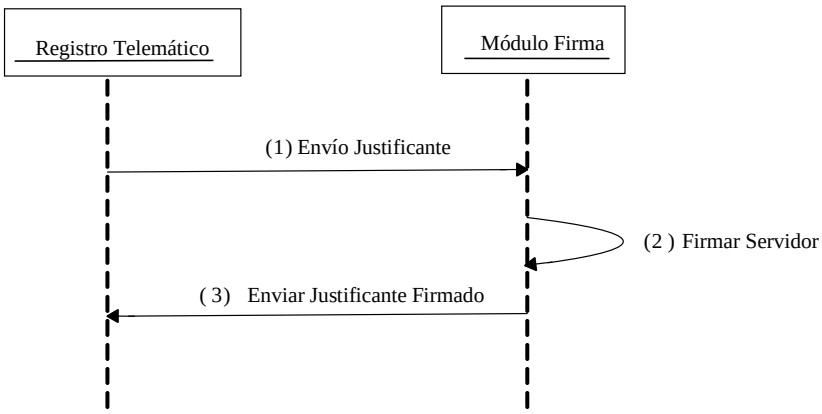

Ilustración 10. Diagrama de Secuencia: Firmar Servidor

- **Envío Justificante**: El módulo Registro Telemático envía el justificante de registro al módulo de firma, para que éste lo firme.
- **Firmar Servidor:** El módulo de firma realiza la firma del justificante de registro.
- **Enviar Justificante Firmado**: El módulo de firma devuelve el justificante de registro firmado.

3.3.6.2 Interfaces con el Módulo de Custodia Documental

A pesar de que el módulo de custodia documental no pertenece al ámbito del proyecto, se incluyen las interfaces que el módulo de registro telemático debe tener con éste ya que son una parte indispensable del funcionamiento del sistema.

3.3.6.2.1 Guardar documento

El módulo de registro telemático enviará al módulo de custodia documental las solicitudes, los justificantes y sus documentos asociados. El módulo de custodia documental devolverá el resultado de la operación y el identificador generado para poder recuperar los documentos.

3.3.6.2.2 Recuperar documento

El módulo de registro telemático solicitará al módulo de custodia documental las solicitudes, los justificantes y sus documentos asociados. El módulo de custodia documental devolverá el resultado de la operación y el identificador generado para poder recuperar los documentos

3.3.6.3 Interfaces con el Módulo de Seguridad

Al igual que el módulo de custodia documental, el módulo de seguridad tampoco pertenece al ámbito del proyecto, pero se incluyen la interfaz por ser parte indispensable para el funcionamiento del sistema.

3.3.6.3.1 Validar ticket

El módulo de registro telemático enviará al módulo de seguridad el ticket que le identifica. El módulo de seguridad comprobará que es válido y devolverá el resultado de la operación.

3.4 Servicio Web de Firma Electrónica

3.4.1 Introducción

La utilización de **medios electrónicos** para la transmisión de datos y documentos ha generado la necesidad de desarrollar instrumentos que otorguen **seguridad jurídica** a las partes que hacen uso de estos medios. La incorporación de las tecnologías de firma electrónica permitirá realizar los trámites administrativos a través de Internet de forma totalmente segura.

Tal y como se explicó previamente, la firma electrónica aporta a los procedimientos electrónicos las cualidades de **autenticidad, confidencialidad, integridad y no repudio** que éstos requieren para proveer la seguridad adecuada en las comunicaciones telemáticas.

La tecnología criptográfica más extendida es aquella que está basada en sistemas de cifrado de clave pública y en la utilización de certificados digitales.

Los certificados son documentos digitales, emitidos por un prestador de servicios de certificación, que dan fe de la vinculación entre una clave pública y una persona física o jurídica.

El certificado digital garantiza la **identidad de las personas físicas o jurídicas**.

Actualmente existe una gran cantidad de plataformas que ofrecen los servicios básicos de firma electrónica: firma de documentos, validación de firmas, validación de certificados, cifrado, etc. El funcionamiento de estas plataformas es en algunos casos totalmente distinto. Sin embargo, todas ellas proporcionan la funcionalidad mínima necesaria para utilizar firma electrónica con todas las garantías jurídicas.

3.4.2 Objetivo

Este módulo proporcionará servicios básicos de firma electrónica independientemente de la plataforma de firma electrónica utilizada (@Firma, ASF, SIAVAL, etc.).

- Validación de certificados
- Creación de firmas electrónicas
- Validación de firmas electrónicas
- Firma en paralelo (dos firmantes firman el mismo documento)
- Firma secuencial (firma sobre otra firma)
- Obtención de información de certificados.

Es importante destacar que estos servicios únicamente se corresponden a **operaciones en servidor**. Para la implementación de firma electrónica en cliente, a través de formularios, será necesario utilizar directamente los componentes ofrecidos por las plataformas de firma electrónica. Estos componentes se basan en applets JAVA o componentes ActiveX.

Se desarrollará un conector que implemente la interfaz definida en el presente documento y se integre con la plataforma de firma electrónica @Firma del Ministerio de Administraciones Públicas.

3.4.3 Contexto

El módulo de firma electrónica estará integrado dentro del un sistema orientado a servicios y proporcionará un servicio horizontal a los restantes módulos.

3.4.4 Diagrama de casos de uso

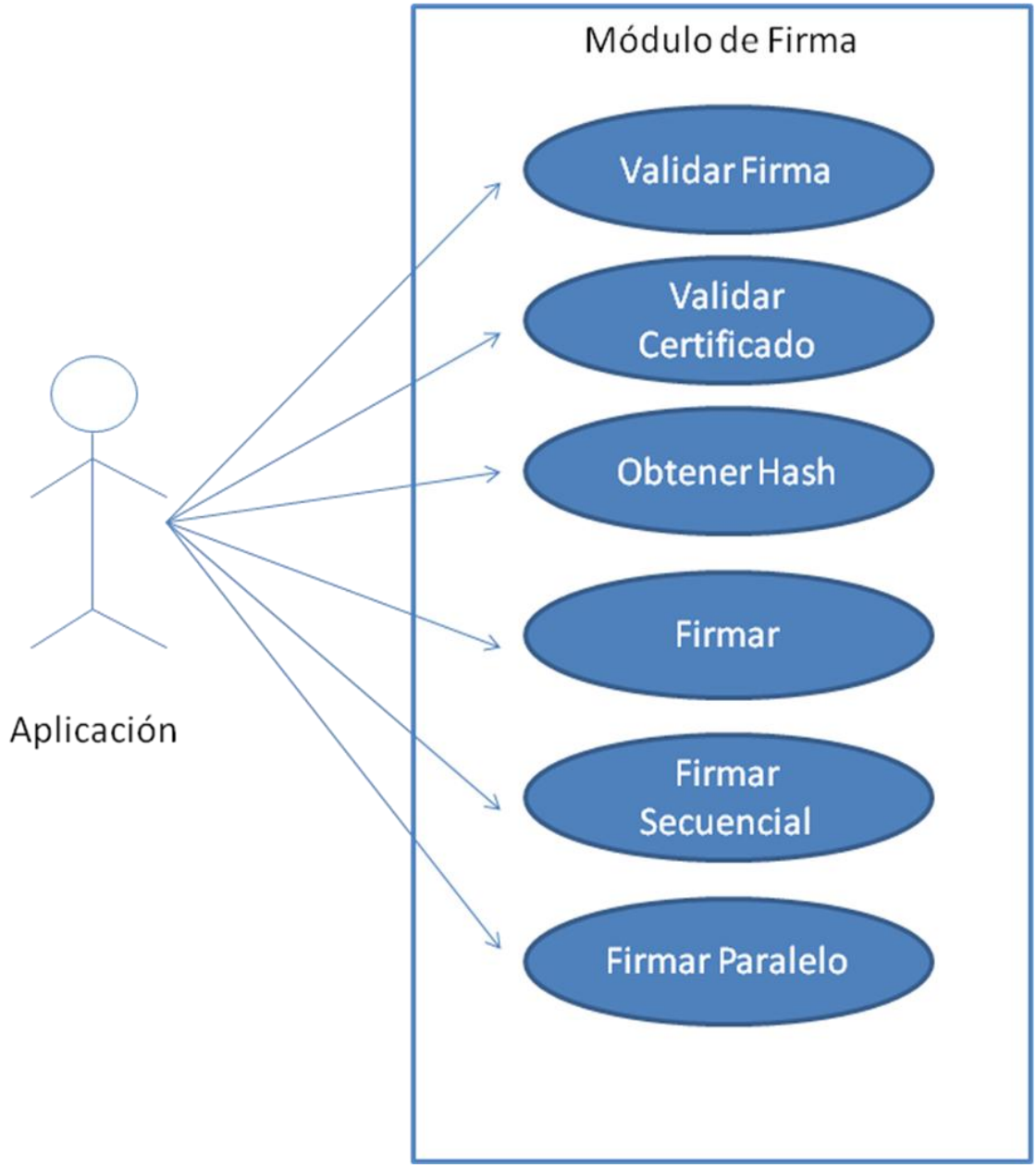

Ilustración 11. Diagrama de casos de uso: Módulo de Firma

3.4.5 Casos de uso

3.4.5.1 Caso de uso Validar Firma

La funcionalidad de validar una firma pretende comprobar si la información recogida en una firma es válida. La validación se solicitará desde un sistema autenticado por el módulo de seguridad, al Servicio de Firma Electrónica.

La validación se podrá hacer en base a la información asociada a la firma: el identificador de la aplicación, el formato de la firma electrónica, el documento no firmado, la propia firma y el ticket que identifica al emisor con la aplicación.

El servicio de firma electrónica validará a la aplicación invocante a través de su identificador y posteriormente mediante el uso de una plataforma de firma, @Firma, se realizará la validación de la misma.

La respuesta del Servicio de Firma Electrónica devolverá el resultado de la validación o una respuesta describiendo el error en caso de no cumplirse las condiciones de validación.

Propósito	Validar una firma electrónica
Precondiciones	Precondiciones: • Disponibilidad plataforma de firma electrónica (@Firma, ASF) Parámetros de entrada: • Identificador de aplicación invocante • Formato de firma electrónica • Documento no firmado • Firma del documento • Ticket que identifica el emisor con la aplicación
Postcondiciones	Resultado del proceso: • Resultado: error o OK • Descripción del resultado: ▪ En caso de error: descripción del error
Circunstancias de uso	• Uso online • Servicio restringido

Tabla 5. Caso de Uso: Validar Firma

3.4.5.1.1 Diagrama de secuencia

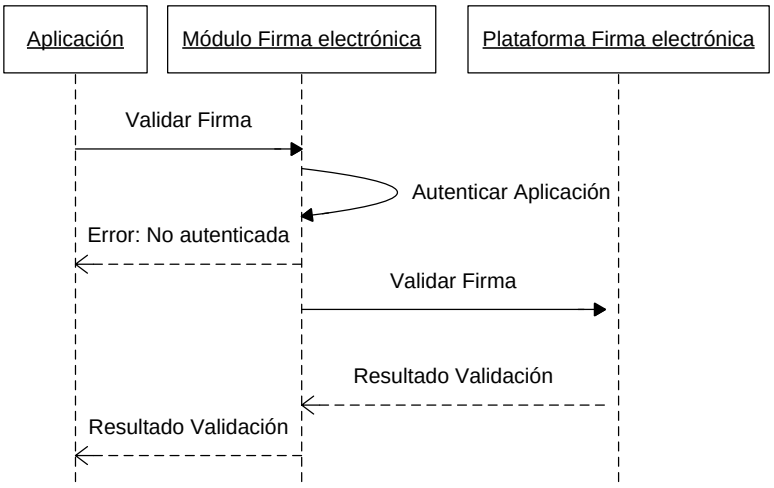

Ilustración 12. Diagrama de Secuencia: Validar Firma

- **Validar Firma**: La Aplicación Usuaria solicita la validación de una firma electrónica mediante el uso de un servicio web del Módulo de Firma Electrónica
- **Valida aplicación**: El Módulo de Firma Electrónica valida el identificador de la aplicación.
- **Resultado de la validación**: Se devuelve un mensaje informando del resultado.
- **Validar Firma**: El Módulo de Firma Electrónica solicita la validación de una firma a la plataforma de firma utilizada, @Firma o ASF.
- **Resultado**: Se devuelve el resultado de la validación.

3.4.5.2 Caso de uso Validar Certificado

Para validar un certificado electrónico hay que comprobar si la información recogida en un certificado es válida. La validación se solicitará desde un sistema autenticado por el módulo de seguridad al Servicio de Firma Electrónica.

La validación se podrá hacer en base a la información asociada al certificado: el identificador de la aplicación, el certificado y el ticket que identifica al emisor con la aplicación.

El servicio de firma electrónica validará a la aplicación invocante a través de su identificador y posteriormente mediante el uso de una plataforma de firma, @Firma, se realizará la validación del certificado.

La respuesta del Servicio de Firma Electrónica devolverá el resultado de la validación o una respuesta describiendo el error en caso de no cumplirse las condiciones de validación.

Propósito	Validar un certificado electrónico (independiente del prestador de servicios: DNIe, FNMT, etc.)
Precondiciones	Precondiciones: • Disponibilidad plataforma de firma electrónica (@Firma, ASF) Parámetros de entrada: • Identificador de aplicación invocante • Certificado • Ticket que identifica el emisor con la aplicación
Postcondiciones	Resultado del proceso: • Resultado de la validación del certificado: ▪ Certificado válido ▪ Certificado caducado ▪ Certificado revocado ▪ Error • Descripción del resultado: ▪ En caso de error: descripción del error
Circunstancias de uso	• Uso online • Servicio restringido

Tabla 6. Caso de Uso: Validar Certificado

3.4.5.2.1 Diagrama de secuencia

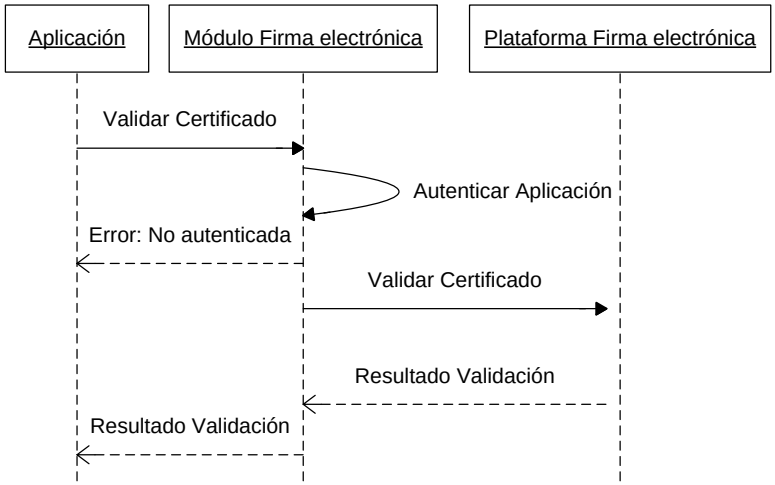

Ilustración 13. Diagrama de secuencia: Validar Certificado

- **Validar Certificado**: La Aplicación Usuaria solicita la validación de un certificado electrónico mediante el uso de un servicio web del Módulo de Firma Electrónica
- **Valida aplicación**: El Módulo de Firma Electrónica valida el identificador de la aplicación.
- **Resultado de la validación**: Se devuelve un mensaje informando del resultado.
- **Validar Certificado**: El Módulo de Firma Electrónica solicita la validación de un certificado a la plataforma de firma utilizada, @Firma o ASF.
- **Resultado**: Se devuelve el resultado de la validación.

3.4.5.3 Caso de uso Obtener Hash

La funcionalidad de obtener un Hash se empleará para recoger la información generada tras la ejecución de un algoritmo matemático. El Hash se solicitará desde un sistema autenticado por el módulo de seguridad, al Servicio de Firma Electrónica.

La Obtención de un Hash se podrá hacer en base a la información asociada: el identificador de la aplicación, el Algoritmo matemático, sus datos y el ticket que identifica al emisor con la aplicación.

El servicio de firma electrónica validará a la aplicación invocante a través de su identificador y posteriormente mediante el uso de una plataforma de firma, @Firma o ASF, se gestionará la obtención del Hash.

La respuesta del Servicio de Firma Electrónica devolverá el Hash o una respuesta describiendo el error en caso de no cumplirse las condiciones de validación.

Propósito	Calcular la hash de un documento
Precondiciones	Precondiciones: • Disponibilidad plataforma de firma electrónica (@Firma, ASF) Parámetros de entrada: • Identificador de aplicación invocante. • Algoritmo • Datos • Ticket que identifica el emisor con la aplicación .
Postcondiciones	Resultado del proceso: • Hash de datos enviados de acuerdo al algoritmo seleccionado • Descripción del resultado: ▪ En caso de error: descripción del error
Circunstancias de uso	• Uso online • Servicio restringido

Tabla 7. Caso de Uso: Obtener Hash

3.4.5.3.1 Diagrama de secuencia

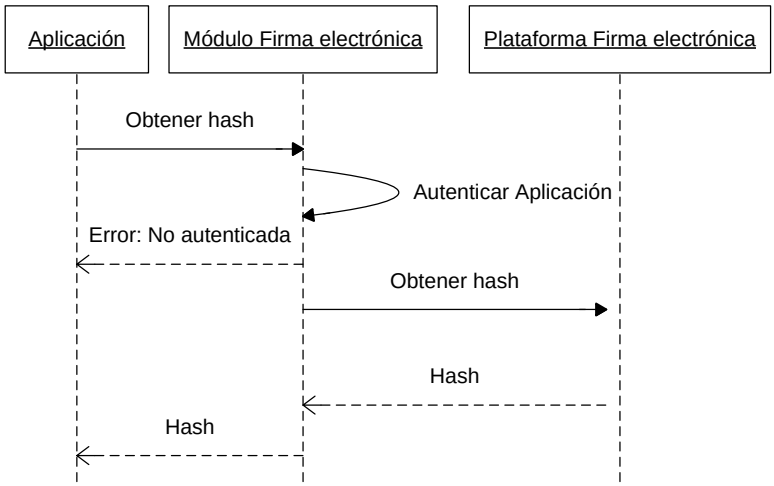

Ilustración 14. Diagrama de Secuencia: Obtener Hash

- **Obtener Hash**: La Aplicación Usuaria solicita el Hash mediante el uso de un servicio web del Módulo de Firma Electrónica
- **Valida aplicación**: El Módulo de Firma Electrónica valida el identificador de la aplicación.
- **Resultado de la validación**: Se devuelve un mensaje informando del resultado.
- **Validar Certificado**: El Módulo de Firma Electrónica solicita el Hash a la plataforma de firma utilizada, @Firma o ASF.
- **Resultado**: Se devuelve el Hash obtenido al aplicar el algoritmo.

3.4.5.4 Caso de uso Firmar

La funcionalidad de firmar en el servidor busca identificar unívocamente al usuario que ha firmado un documento. La firma se solicitará desde un sistema autenticado por el módulo de seguridad, al Servicio de Firma Electrónica.

Este servicio se podrá hacer en base a la información asociada a la firma: el identificador de la aplicación, el formato de la firma electrónica, el nombre del documento, el propio documento, el tipo de documento, el formato de la firma y el ticket que identifica al emisor con la aplicación.

El servicio de firma electrónica validará a la aplicación invocante a través de su identificador y posteriormente mediante el uso de una plataforma de firma, @Firma o ASF, se realizará la firma del documento.

La respuesta del Servicio de Firma Electrónica devolverá como resultado la firma y el identificador de la transición, que será útil en caso de utilizar, posteriormente, los servicios de firma secuencial o firma en paralelo, o una respuesta describiendo el error si no se cumplen las condiciones de validación.

Propósito	Firma de un documento
Precondiciones	Precondiciones: • Disponibilidad plataforma de firma electrónica (@Firma, ASF) Parámetros de entrada: • Identificador de aplicación invocante. • Formato de firma • Nombre documento • Tipo documento • Documento • Ticket que identifica el emisor con la aplicación.
Postcondiciones	Resultado del proceso: • Resultado • Descripción del resultado: En caso de error: descripción del error. • ID de transacción • Firma
Circunstancias de uso	• Uso online • Servicio restringido

Tabla 8. Caso de Uso: Firmar

3.4.5.4.1 Diagrama de secuencia

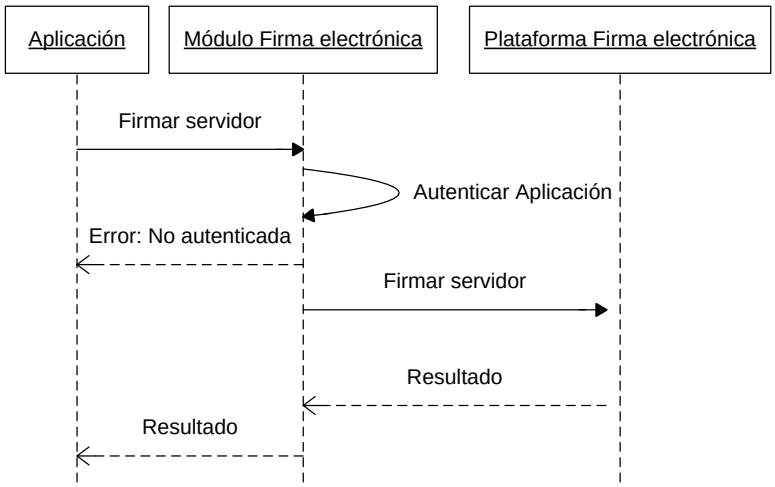

Ilustración 15. Diagrama de secuencia: Firmar servidor

- **Firmar en el Servidor**: La Aplicación Usuaria solicita la firma de un documento mediante el uso de un servicio web del Módulo de Firma Electrónica
- **Valida aplicación**: El Módulo de Firma Electrónica valida el identificador de la aplicación.
- **Resultado de la validación**: Se devuelve un mensaje informando del resultado.
- **Firmar en el Servidor**: El Módulo de Firma Electrónica solicita la firma de un documento a la plataforma de firma utilizada, @Firma o ASF.
- **Resultado**: Se devuelve la firma y un identificador de la transición.

3.4.5.5 Caso de uso Firmar secuencial

Su objetivo es permitir la firma secuencial de distintos firmantes de forma que se genere un único archivo de firmas con todos los firmantes y no múltiples firmas individuales de un mismo archivo. La firma secuencial se solicitará desde un sistema autenticado por el módulo de seguridad, al Servicio de Firma Electrónica.

Este servicio se podrá hacer en base a la información asociada a la firma secuencial: el identificador de la aplicación, el número de transacción, el identificador de la transacción y el ticket que identifica al emisor con la aplicación.

El servicio de firma electrónica validará a la aplicación invocante a través de su identificador y posteriormente mediante el uso de una plataforma de firma, @Firma o ASF, se realizará la firma del documento.

La respuesta del Servicio de Firma Electrónica devolverá como resultado la firma y el identificador de la transición, que será útil en caso de utilizar, posteriormente, los servicios de

firma secuencial o en paralelo, o una respuesta describiendo el error si no se cumplen las condiciones de validación.

Propósito	Firma secuencial de un documento: una aplicación firma sobre la firma de un documento.
Precondiciones	• Disponibilidad plataforma de firma electrónica (@Firma, ASF, etc.) • Disponer del nº de transacción Parámetros de entrada: • Identificador de aplicación invocante • ID transacción • Ticket que identifica el emisor con la aplicación.
Postcondiciones	Resultado del proceso: • Resultado • Descripción del resultado: En caso de error: descripción del error • ID de transacción • Firma
Circunstancias de uso	• Uso online • Servicio restringido

Tabla 9. Caso de Uso: Firma Secuencial

3.4.5.5.1 Diagrama de secuencia

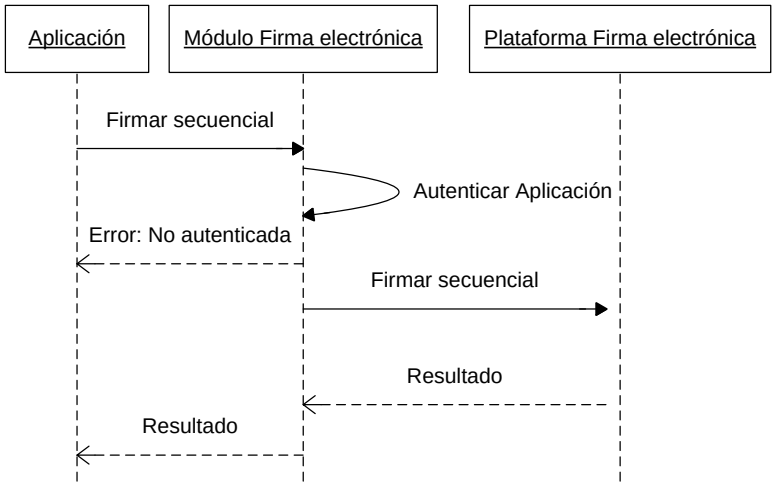

Ilustración 16. Diagrama de Secuencia: Firma Secuencial

- **Firma secuencial**: La Aplicación Usuaria solicita la firma secuencial de un documento mediante el uso de un servicio web del Módulo de Firma Electrónica
- **Valida aplicación**: El Módulo de Firma Electrónica valida el identificador de la aplicación.
- **Resultado de la validación**: Se devuelve un mensaje informando del resultado.
- **Firma secuencial**: El Módulo de Firma Electrónica solicita la firma secuencial de un documento a la plataforma de firma utilizada, @Firma o ASF.
- **Resultado**: Se devuelve la firma y un identificador de la transición.

3.4.5.6 Caso de uso Firmar paralelo

Su objetivo es permitir la firma en paralelo, es decir, simultáneamente, de distintos firmantes, de forma que se genere un único archivo de firmas con todos los firmantes y no múltiples firmas individuales de un mismo archivo. La firma en paralelo se solicitará desde un sistema autenticado por el módulo de seguridad al Servicio de Firma Electrónica.

Este servicio se podrá hacer en base a la información asociada a la firma en paralelo: el identificador de la aplicación, el número de transacción, el identificador de la transacción y el ticket que identifica al emisor con la aplicación.

El servicio de firma electrónica validará a la aplicación invocante a través de su identificador y posteriormente mediante el uso de una plataforma de firma, @Firma o ASF, se realizará la firma del documento.

La respuesta del Servicio de Firma Electrónica devolverá como resultado la firma y el identificador de la transición, que será útil en caso de utilizar, posteriormente, los servicios de firma secuencial o en paralelo, o una respuesta describiendo el error si no se cumplen las condiciones de validación.

Propósito	Una aplicación firma un documento que ya contiene una firma
Precondiciones	• Disponibilidad plataforma de firma electrónica (@Firma, ASF, etc.) • Disponer del nº de transacción Parámetros de entrada: • Identificador de aplicación invocante • ID transacción • Ticket que identifica el emisor con la aplicación
Postcondiciones	Resultado del proceso: • Resultado • Descripción del resultado: En caso de error: descripción del error. • ID de transacción • Firma
Circunstancias de uso	• Uso online • Servicio restringido

Tabla 10. Caso de Uso: Envío de Nueva Solicitud

3.4.5.6.1 Diagrama de secuencia

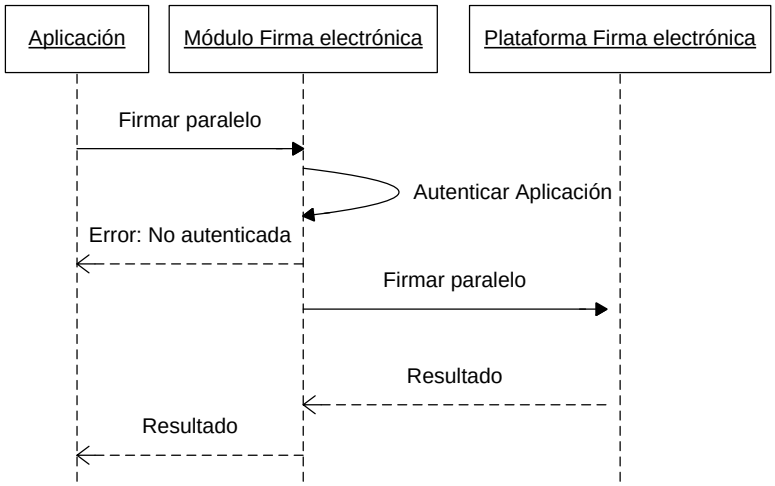

Ilustración 17. Diagrama de Secuencia: Firmar Paralelo

- **Firma en paralelo**: La Aplicación Usuaria solicita la firma en paralelo de un documento mediante el uso de un servicio web del Módulo de Firma Electrónica
- **Valida aplicación**: El Módulo de Firma Electrónica valida el identificador de la aplicación.
- **Resultado de la validación**: Se devuelve un mensaje informando del resultado.
- **Firma en paralelo**: El Módulo de Firma Electrónica solicita la firma en paralelo secuencial de un documento a la plataforma de firma utilizada, @Firma o ASF.
- **Resultado**: Se devuelve la firma y un identificador de la transición.

4. ARQUITECTURA E IMPLEMENTACIÓN

El objetivo primordial del apartado de Diseño es el de resolver el problema descrito y modelado en el Análisis del Sistema. Para resolver dicho problema se definirá la arquitectura del sistema y del entorno tecnológico que le va a dar soporte, junto con la especificación detallada de los componentes del sistema de información.

4.1 Diseño del sistema

El diseño del sistema se ocupa de desarrollar las directrices propuestas durante el análisis en función de aquella configuración que tenga más posibilidades de satisfacer los objetivos planteados, tanto desde el punto de vista funcional (requisitos del usuario) como del no funcional (rendimiento, plataforma, utilización de recursos).

Así, las principales **características** del sistema que se va a implementar son:

- Plataforma tecnológica puntera orientada a facilitar y hacer más **rápida y flexible** la provisión y **prestación de servicios** públicos electrónicos.
- Solución de **software libre**.
- Basada en **estándares**. Para el desarrollo de la arquitectura se han tenido en cuenta los patrones de núcleo **J2EE** e incorporado los más importantes estándares del mercado: Spring, Hibernate, Xfire, etc.
- **Arquitectura SOA**
- **Solución no intrusiva** gracias a una arquitectura de conectores.

En los próximos apartados se definirá la arquitectura y las tecnologías que se emplearán en la implementación.

4.1.1 Arquitectura

Se define una **Arquitectura Software** de la siguiente manera:

- Consiste en un conjunto de **patrones y abstracciones coherentes** que proporcionan el **marco de referencia** necesario para guiar la construcción del software para un sistema de información.
- Establece los fundamentos para que analistas, diseñadores, programadores, etc. trabajen en una **línea común** que permita alcanzar los objetivos del sistema de información, cubriendo todas las necesidades.
- **Define**, de manera abstracta, los **componentes** que llevan a cabo alguna tarea de computación, sus **interfaces** y la **comunicación** ente ellos.

Generalmente, no es necesario inventar una nueva arquitectura de software para cada sistema de información. Lo habitual es adoptar una arquitectura conocida en función de sus ventajas e inconvenientes para cada caso en concreto. Así, las arquitecturas más comunes son:

- Monolítica. Donde el software se estructura en grupos funcionales muy acoplados.
- Cliente-servidor. Donde el software reparte su carga de cómputo en dos partes independientes pero sin reparto claro de funciones.
- Arquitectura de tres niveles. Especialización de la arquitectura cliente-servidor donde la carga se divide en tres partes con un reparto claro de funciones: una capa para la presentación, que corresponde a la interfaz de usuario; otra para el cálculo, donde se encuentra modelado el negocio; y otra para el almacenamiento, también denominado persistencia. Una capa solamente tiene relación con la siguiente.

Sin embargo, actualmente existe una tendencia a **implantar arquitecturas de desarrollo** que proporcionen una **base tecnológica** fiable, moderna y segura que sirva como pilar del desarrollo de nuevos sistemas de información.

Por ello, se ha escogido como solución para implementar este proyecto la **arquitectura orientada a servicios**. En este tipo de arquitecturas, la aplicación está formada por servicios débilmente acoplados pero altamente interoperables. Esto quiere decir, que pueden relacionarse entre sí, pero son independientes.

Estos servicios se comunican unos con otros mediante mensajes independientes del lenguaje de programación y de la arquitectura hardware del sistema. De esta manera, sólo es necesario definir las interfaces de cada servicio. En este caso, al igual que en un gran número de arquitecturas SOA, los mensajes siguen el protocolo WSDL.

Los beneficios que puede obtener una aplicación con arquitectura SOA son:

- Mejora en los tiempos de realización de **cambios en procesos**.
- Facilidad para abordar modelos de negocios basados en **colaboración con otros entes** (socios, proveedores).
- Capacidad para **reemplazar elementos** de la capa aplicativa SOA **sin interrumpir el funcionamiento** de la aplicación.
- Facilidad para la **integración de tecnologías no relacionadas** entre sí.

En la siguiente ilustración se puede observar la arquitectura del sistema, así como los servicios concretos que abarca este proyecto.

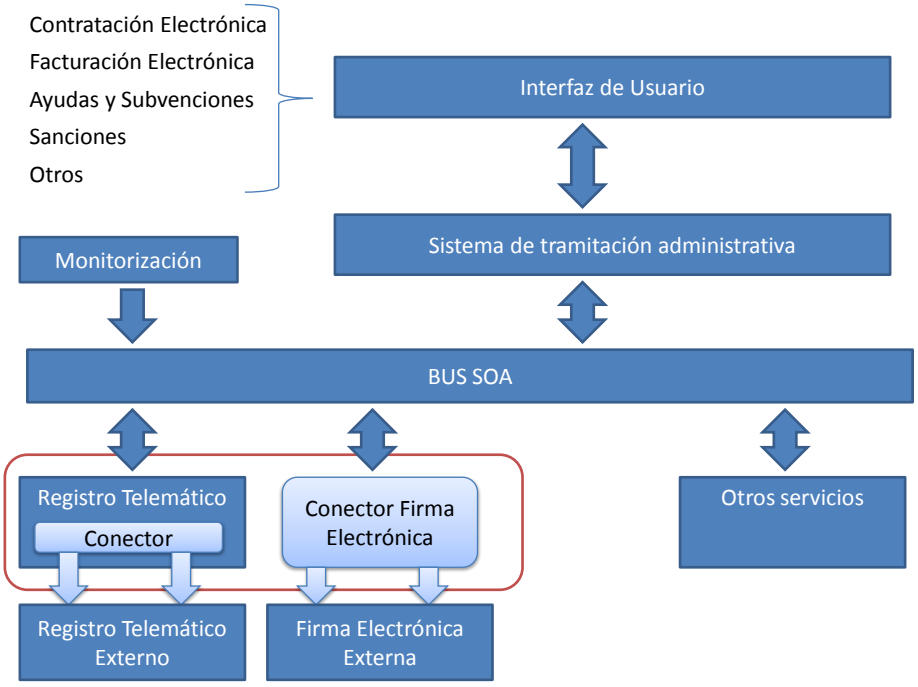

Ilustración 18. Arquitectura Funcional

4.2 Pautas generales

Los módulos estarán divididos en los siguientes proyectos:

- Api: Almacena el código fuente java de cada módulo, así como sus pruebas unitarias
- Web: Almacena los recursos web del módulo.
- Wsappclient: Es el encargado de publicar un módulo como servicio web.

La estructura de cada proyecto será la siguiente:

src

main

java (Carpeta donde se almacenará todo el código Java)

resources (Carpeta donde se almacenarán los ficheros hbm.xml, los ficheros de configuración de servicios, ficheros de configuración de servicios, y en general recursos que se desee que sean visibles en el *classpath*)

test

java (Carpeta donde se almacenará el código Java que sirva para probar unitariamente los desarrollos. Una prueba deberá residir en el mismo paquete que la clase que prueba, y llamarse igual, poniendo la cadena Test al final. Por ejemplo: es.paquete.PruebaTest probará es.paquete.Prueba)

webapp (Carpeta donde se almacenarán los recursos web)

pom.xml (Fichero que contiene las dependencias del proyecto con otras bibliotecas y proyectos)

4.2.1 Control de errores

Se realizará control de errores en varias capas:

- Control de errores en la capa de presentación: se propaga la excepción.
- Control de errores en la capa de acceso a datos: se escribe el error y se propaga.
- Control de errores en la capa de acceso a datos: se escribe el error y se propaga.

4.2.2 Sesión en los DAO

Una vez obtenida la sesión, debe asegurarse mediante código fuente que ésta no puede permanecer bloqueada.

4.2.3 Campos estándar de la base de datos

Todas las entidades poseerán unos campos "internos", es decir, controlados por la aplicación, que no podrán ser modificados por el usuario

- id: identificador del registro que se creará mediante la herramienta Hibernate, con una secuencia.
- versión: versión del registro que se actualizará mediante la herramienta Hibernate.
- idUsuarioCreacion: Identificador del usuario que crea el registro, una vez creado no se permitirá su modificación.
- idUsuarioModificacion: Identificador del usuario de modificación, se actualizará cada vez que se modifique o elimine el registro.

- idEstadoRegistro Identificador lógico del estado del registro (1 alta, 0 baja).
- fechaCreacion: Fecha de creación del registro, una vez creado no se permitirá su modificación.
- fechaModificacion: Fecha de modificación, se actualizará con la fecha del sistema cada vez que se modifique o elimine el registro.

4.3 Diseño Detallado

4.3.1 Registro Telemático

Este apartado recoge el diseño técnico del módulo de Registro Telemático.

Está dividido en dos partes principales:

- Interfaz
  - Se describe de forma detallada cada uno de los servicios que prestará la interfaz.
- Conectores
  - Se describen de forma detallada los conectores que implementan el interfaz definido.

4.3.1.1 Diseño detallado del interfaz

El interfaz del módulo de registro telemático define los siguientes servicios web, que corresponden a los casos de uso especificados en el modelo:

- Envío Nueva Solicitud
- Búsqueda Registros
- Consulta Registro
- Eliminación Registro

4.3.1.1.1 Servicio Envío Nueva Solicitud

A través de este servicio se solicitará al Registro Telemático una nueva solicitud de registro.

PARÁMETROS DE ENTRADA		
NOMBRE	TIPO	DESCRIPCIÓN
idAplicacion	String	Identificador de la aplicación. Deberá ser dado de alta en el módulo de Seguridad
solicitud	String	XML con los datos genéricos y específicos de la solicitud
anexos	Array	Array de documentos con la siguiente estructura: • código (String) • tipoDocumento (String) • nombre (String) • contenido (String base 64) • hash (String base 64) • algoritmoHash (String)
formatoFirma	String	El formato de firma utilizado para firmar la solicitud.
ticket	String	Ticket que identifica el emisor con la aplicación

Tabla 11. Servicio Envío Nueva Solicitud: Parámetros de Entrada

PARÁMETROS DE SALIDA		
NOMBRE	TIPO	DESCRIPCIÓN
resultado	Bool	Posibles valores: • TRUE en caso de ejecución correcta • FALSE en caso de producirse un error
justificante	String	Datos del justificante de registro
descripción	String	En caso de que error = FALSE • Vacío En caso de que error = TRUE • Descripción del error

Tabla 12. Servicio Envío Nueva Solicitud: Parámetros de Salida

CÓDIGOS DE ERROR	
CÓDIGO	DESCRIPCIÓN
ERR_ACCESO	Aplicación no válida
ERR_CONEX	Error de conexión con plataforma de registro telemático
ERR_FIRMA	La firma de la documentación no es correcta y no se inserta la petición
ERR_HASH	El hash de alguno de los anexos no es correcto
ERR_FORMATO	Los datos son incorrectos para Registro Telemático. No se inserta la petición
ERR_BBDD	Error en el acceso a base de datos

Tabla 13. Servicio Envío Nueva Solicitud: Códigos de Error

4.3.1.1.2 Servicio Búsqueda Registros

El servicio de búsqueda de registros proporciona un interfaz para poder obtener un listado de los registros de acuerdo a los criterios de búsqueda seleccionados.

PARÁMETROS DE ENTRADA		
NOMBRE	TIPO	DESCRIPCIÓN
idAplicacion	String	Identificador de la aplicación. Deberá ser dado de alta en el módulo de Seguridad
idRegistro	String	Identificador del Registro
nifRemitente	String	NIF del Remitente
nombreRemitente	String	Nombre del Remitente
fechaRecepciónDesde	String	Fecha de Recepción desde
fechaRecepciónHasta	String	Fecha de Recepción hasta
codDestino	String	Código del Destino del Registro
codAsunto	String	Código del Asunto del Registro
ticket	String	Ticket que identifica el emisor con la aplicación

Tabla 14. Servicio Búsqueda de Registros: Parámetros de Entrada

PARÁMETROS DE SALIDA		
NOMBRE	TIPO	DESCRIPCIÓN
resultado	Bool	Posibles valores: • TRUE en caso de ejecución correcta • FALSE en caso de producirse un error
descripción	String	En caso de que error = FALSE • XML con el listado de registros: ▪ idRegistro ▪ NIFRemitente ▪ NombreRemitente ▪ fechaRecepcion ▪ codDestino ▪ codAsunto ▪ codAplicacion En caso de que error = TRUE • Descripción del error

Tabla 15. Servicio Búsqueda de Registros: Parámetros de Salida

CÓDIGOS DE ERROR	
CÓDIGO	DESCRIPCIÓN
ERR_ACCESO	Aplicación no válida
ERR_CONEX	Error de conexión con plataforma de registro telemático
ERR_FILTRO	No se puede consultar. Los datos del filtro no son correctos
ERR_BBDD	Error en el acceso a base de datos

Tabla 16. Servicio Búsqueda de Registros: Códigos de Error

4.3.1.1.3 Servicio Consulta Registro

El servicio de consulta de registro proporciona un interfaz para poder consultar el detalle de un registro.

PARÁMETROS DE ENTRADA		
NOMBRE	TIPO	DESCRIPCIÓN
idAplicacion	String	Identificador de la aplicación. Deberá ser dado de alta en el módulo de Seguridad
idRegistro	String	Identificador del Registro.
Ticket	String	Ticket que identifica el emisor con la aplicación

Tabla 17. Servicio Consulta de Registros: Parámetros de Entrada

PARÁMETROS DE SALIDA		
NOMBRE	TIPO	DESCRIPCIÓN
resultado	Bool	Posibles valores: • TRUE en caso de ejecución correcta • FALSE en caso de producirse un error
descripción	String	En caso de que error = FALSE • XML con el detalle del registro: ▪ idRegistro ▪ NIFRemitente ▪ NombreRemitente ▪ fechaRecepcion ▪ codDestino ▪ codAsunto ▪ codAplicacion ▪ solicitud ▪ anexos ▪ justificante En caso de que error = TRUE • Descripción del error

Tabla 18. Servicio Consulta de Registros: Parámetros de Salida

CÓDIGOS DE ERROR	
CÓDIGO	DESCRIPCIÓN
ERR_ACCESO	Aplicación no válida
ERR_CONEX	Error de conexión con plataforma de registro telemático
ERR_FILTRO	No se puede consultar. Los datos del filtro no son correctos
ERR_BBDD	Error en el acceso a base de datos

Tabla 19. Servicio Consulta de Registros: Códigos de Error

4.3.1.1.4 Servicio Eliminación Registro

El servicio de eliminación de registro proporciona un interfaz para poder eliminar un registro concreto del repositorio de registro telemático.

PARÁMETROS DE ENTRADA		
NOMBRE	TIPO	DESCRIPCIÓN
idAplicacion	String	Identificador de la aplicación. Deberá ser dado de alta en el módulo de Seguridad
idRegistro	String	Identificador del Registro
ticket	String	Ticket que identifica el emisor con la aplicación

Tabla 20. Servicio Eliminación de Registros: Parámetros de Entrada

PARÁMETROS DE SALIDA		
NOMBRE	TIPO	DESCRIPCIÓN
resultado	Bool	Posibles valores: • TRUE en caso de ejecución correcta • FALSE en caso de producirse un error
descripción	String	En caso de que error = FALSE • Vacío En caso de que error = TRUE • Descripción del error

Tabla 21. Servicio Eliminación de Registros: Parámetros de Salida

CÓDIGOS DE ERROR	
CÓDIGO	DESCRIPCIÓN
ERR_ACCESO	Aplicación no válida
ERR_CONEX	Error de conexión con plataforma de registro telemático
ERR_ELIMINAR	El proceso de eliminación de registros no es correcto
ERR_BBDD	Error en el acceso a base de datos

Tabla 22. Servicio Eliminación de Registros: Códigos de Error

4.3.1.2 Diseño detallado del conector

4.3.1.2.1 Introducción

El conector simplemente actúa como envoltorio, es decir, el registro telemático implementa exactamente la misma interfaz. La finalidad del conector es independizar la conexión entre el bus de datos y el registro telemático. De esta forma, si se desea emplear un registro telemático distinto, tan sólo es necesario adaptarlo a la interfaz del conector.

4.3.1.3 Diseño detallado del módulo registro telemático

4.3.1.3.1 Solicitud de registro

La solicitud de registro será un documento XML con las siguientes partes:

- Datos descriptivos de la solicitud de registro:
  - Tipo
  - Versión del documento
  - Tipo de registro: entrada / salida
- Datos Genéricos: datos comunes a cualquier tipo de solicitud.
  - Código destino
  - Código de asunto
  - Código de oficina
  - NIF remitente
  - NIF destinatario
  - Nombre remitente
  - Número expediente
  - Fecha recepción
- Datos específicos: dependientes de cada aplicación
- Documentos anexos: tipo de dato complejo con la siguiente estructura:
  - Documentos
    - Código documento
    - Rol
    - Nombre documento
    - Hash documento
    - Algoritmo Hash

La solicitud se firmará en formato XAdES-BES de forma previa a proceder a su registro.

4.3.1.3.2 Justificante de registro

El justificante de registro deberá contener, en uno de sus campos, la solicitud XML y deberá firmarse en formato XAdES-BES.

4.3.1.3.3 Modelo de datos

El modelo de datos se representa empleando la nomenclatura propia de MySql. El gráfico que corresponde a dicho modelo es el siguiente:

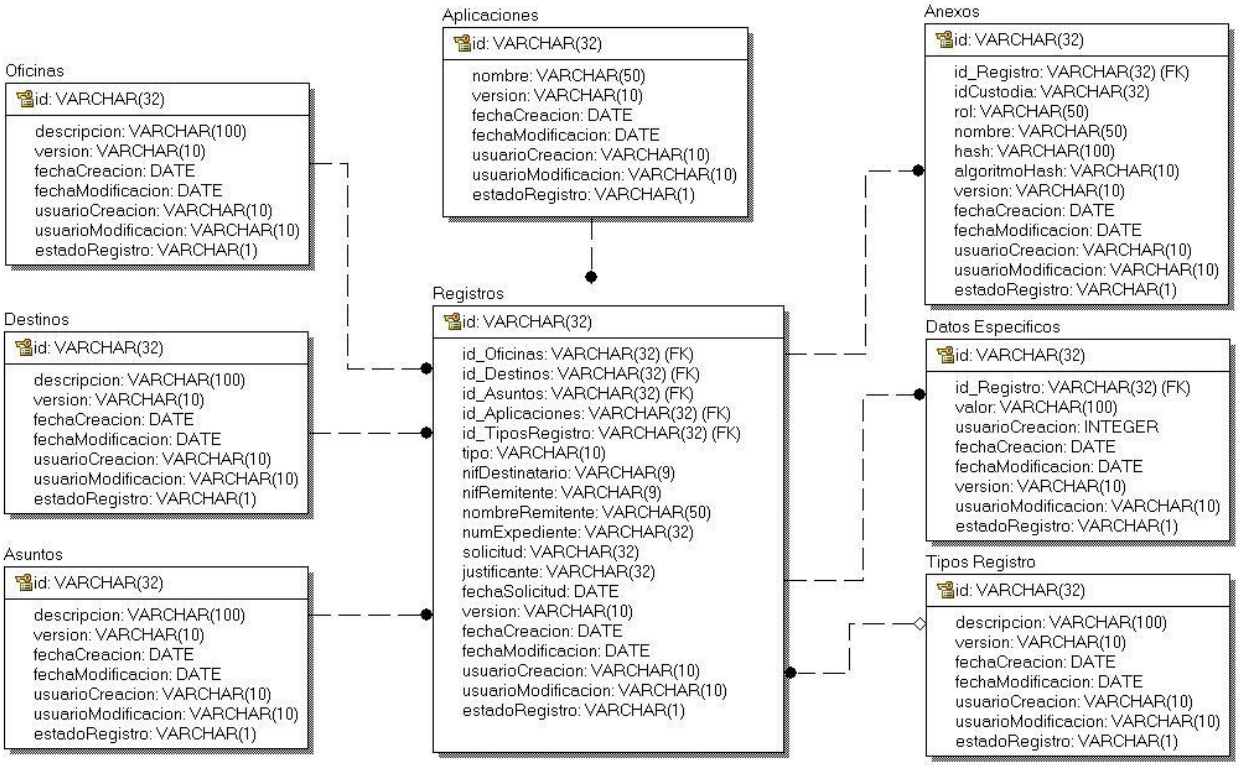

Ilustración 19. Modelo de Datos

TABLA: REGISTROS		
COLUMNA	TIPO	DESCRIPCIÓN
id	VARCHAR(32)	• No nulo • Clave primaria
tipo	VARCHAR	Tipo de la solicitud • Entrada • Salida
nifDestinatario	VARCHAR(9)	NIF del destinatario de la solicitud
nifRemitente	VARCHAR(9)	NIF del remitente de la solicitud
nombreRemitente	VARCHAR(50)	Nombre del remitente de la solicitud
numExpediente	VARCHAR(32)	Número de expediente asociado al registro
fechaSolicitud	Datetime	Fecha de la solicitud
solicitud	VARCHAR(32)	Identificador del archivo de la solicitud en el módulo de custodia documental
justificante	VARCHAR(32)	Identificador del archivo del justificante en el módulo de custodia documental
versión	VARCHAR	Número de la versión del registro
usuarioCreación	VARCHAR	Usuario que crea el registro
usuarioModificación	VARCHAR	Usuario que modifica el registro
fechaRecepcion	Datetime	• No nulo • Fecha de registro
fechaModificación	Datetime	Fecha de modificación del registro
estadoRegistro	VARCHAR(1)	Estado del registro • ALTA • ELIMINADO
DESTINOS_idDESTINOS (FK)	VARCHAR(32)	Código de destino
ASUNTOS_idASUNTOS (FK)	VARCHAR(32)	Código de asunto
OFICINAS_idOFICINAS (FK)	VARCHAR(32)	Código de oficina
APLICACIONES_idAPLICACIONES	VARCHAR(32)	ID de aplicación que invoca al servicio web
TIPOS_REGISTRO_idTIPOS_REGISTRO	VARCHAR(32)	Código de tipo de registro

Tabla 23. Modelo de Datos: Registros

TABLA: DESTINOS		
COLUMNA	TIPO	DESCRIPCIÓN
id	VARCHAR(32)	• No nulo • Clave primaria
descripción	VARCHAR(100)	Descripción del destino
versión	VARCHAR	Número de la versión del registro
usuarioCreación	VARCHAR	Usuario que crea el registro
usuarioModificación	VARCHAR	Usuario que modifica el registro
fechaRecepcion	Datetime	• No nulo • Fecha de registro
fechaModificación	Datetime	Fecha de modificación
estadoRegistro	VARCHAR(1)	Estado del registro • ALTA • ELIMINADO

Tabla 24. Modelo de Datos: Destinos

TABLA: ASUNTOS		
COLUMNA	TIPO	DESCRIPCIÓN
id	VARCHAR(32)	• No nulo • Clave primaria
descripción	VARCHAR(100)	Descripción del asunto
versión	VARCHAR	Número de la versión del registro
usuarioCreación	VARCHAR	Usuario que crea el registro
usuarioModificación	VARCHAR	Usuario que modifica el registro
fechaRecepcion	Datetime	• No nulo • Fecha de registro
fechaModificación	Datetime	Fecha de modificación del registro
estadoRegistro	VARCHAR(1)	Estado del registro • ALTA • ELIMINADO

Tabla 25. Modelo de Datos: Asuntos

TABLA: APLICACIONES		
COLUMNA	TIPO	DESCRIPCIÓN
id	VARCHAR(32)	• No nulo • Clave primaria
descripción	VARCHAR(100)	Descripción de la aplicación
versión	VARCHAR	Número de la versión del registro
usuarioCreación	VARCHAR	Usuario que crea el registro
usuarioModificación	VARCHAR	Usuario que modifica el registro
fechaRecepcion	Datetime	• No nulo • Fecha de registro
fechaModificación	Datetime	Fecha de modificación del registro
estadoRegistro	VARCHAR(1)	Estado del registro • ALTA • ELIMINADO

Tabla 26. Modelo de Datos: Aplicaciones

TABLA: OFICINAS		
COLUMNA	TIPO	DESCRIPCIÓN
id	VARCHAR(32)	• No nulo • Clave primaria
descripción	VARCHAR(100)	Descripción de la oficina
versión	VARCHAR	Número de la versión del registro
usuarioCreación	VARCHAR	Usuario que crea el registro
usuarioModificación	VARCHAR	Usuario que modifica el registro
fechaRecepcion	Datetime	• No nulo • Fecha de registro
fechaModificación	Datetime	Fecha de modificación del registro
estadoRegistro	VARCHAR(1)	Estado del registro • ALTA • ELIMINADO

Tabla 27. Modelo de Datos: Oficinas

TABLA: TIPOS REGISTROS		
COLUMNA	TIPO	DESCRIPCIÓN
id	VARCHAR(32)	• No nulo • Clave primaria
descripción	VARCHAR(100)	Descripción del tipo especifico
versión	VARCHAR	Número de la versión del registro
usuarioCreación	VARCHAR	Usuario que crea el registro
usuarioModificación	VARCHAR	Usuario que modifica el registro
fechaRecepcion	Datetime	• No nulo • Fecha de registro
fechaModificación	Datetime	Fecha de modificación del registro
estadoRegistro	VARCHAR(1)	Estado del registro • ALTA • ELIMINADO

Tabla 28. Modelo de Datos: Tipos Registros

TABLA: DATOS ESPECIFICOS		
COLUMNA	TIPO	DESCRIPCIÓN
id	VARCHAR(32)	• No nulo • Clave primaria
valor	VARCHAR(100)	Valor del dato específico
versión	VARCHAR	Número de la versión del registro
usuarioCreación	VARCHAR	Usuario que crea el registro
usuarioModificación	VARCHAR	Usuario que modifica el registro
fechaRecepcion	Datetime	• No nulo • Fecha de registro
fechaModificación	Datetime	Fecha de modificación del registro
estadoRegistro	VARCHAR(1)	Estado del registro • ALTA • ELIMINADO

Tabla 29. Modelo de Datos: Datos Específicos

TABLA: ANEXOS		
COLUMNA	TIPO	DESCRIPCIÓN
id	VARCHAR(32)	• No nulo • Clave primaria
nombre	VARCHAR(50)	Nombre del documento
codCustodia	VARCHAR(32)	Código del documento en el módulo de custodia
rol	VARCHAR	Rol del documento
hash	VARCHAR(100)	Función resumen del documento
algoritmoHash	VARCHAR(10)	Algoritmo empleado para obtener la función hash
versión	VARCHAR	Número de la versión del registro
usuarioCreación	VARCHAR	Usuario que crea el registro
usuarioModificación	VARCHAR	Usuario que modifica el registro
fechaRecepcion	Datetime	• No nulo • Fecha de registro
fechaModificación	Datetime	Fecha de modificación del registro
estadoRegistro	VARCHAR(1)	Estado del registro • ALTA • ELIMINADO
REGISTROS_idREGISTROS (FK)	VARCHAR(32)	Código de registro

Tabla 30. Modelo de Datos: Anexos

4.3.1.3.4 Procesos

4.3.1.3.4.1 Envío nueva solicitud

PARÁMETROS DE ENTRADA		
NOMBRE	TIPO	DESCRIPCIÓN
idAplicacion	String	Tipo: **requerido** Identificador de la aplicación
ticket	String	Tipo: **requerido** Ticket de control para gestionar la seguridad. Será proporcionado a las aplicaciones por el módulo de seguridad
solicitud	String	Tipo: **requerido** Solicitud XML firmada
tipo	String	Tipo: **requerido** Tipo de registro: • Entrada • Salida
anexos	Array	Tipo: **opcional** Array de documentos con la siguiente estructura: • código (String) • nombre (String) • contenido (String base 64) • hash (String base 64) • algoritmoHash (String)
formatoFirma	String	Formato de firma: por defecto XAdES-BES

Tabla 31. Proceso de Envío Nueva Solicitud. Parámetros de Entrada

PARÁMETROS DE SALIDA		
NOMBRE	TIPO	DESCRIPCIÓN
resultado	Bool	Posibles valores: • TRUE en caso de ejecución correcta • FALSE en caso de producirse un error
justificante	String	Justificante de registro en formato XML.
descripción	String	En caso de que error = FALSE • Vacío. En caso de que error = TRUE • Descripción del error.

Tabla 32. Proceso de Envío Nueva Solicitud. Parámetros de Salida

4.3.1.3.4.1.1 Diagrama de secuencia

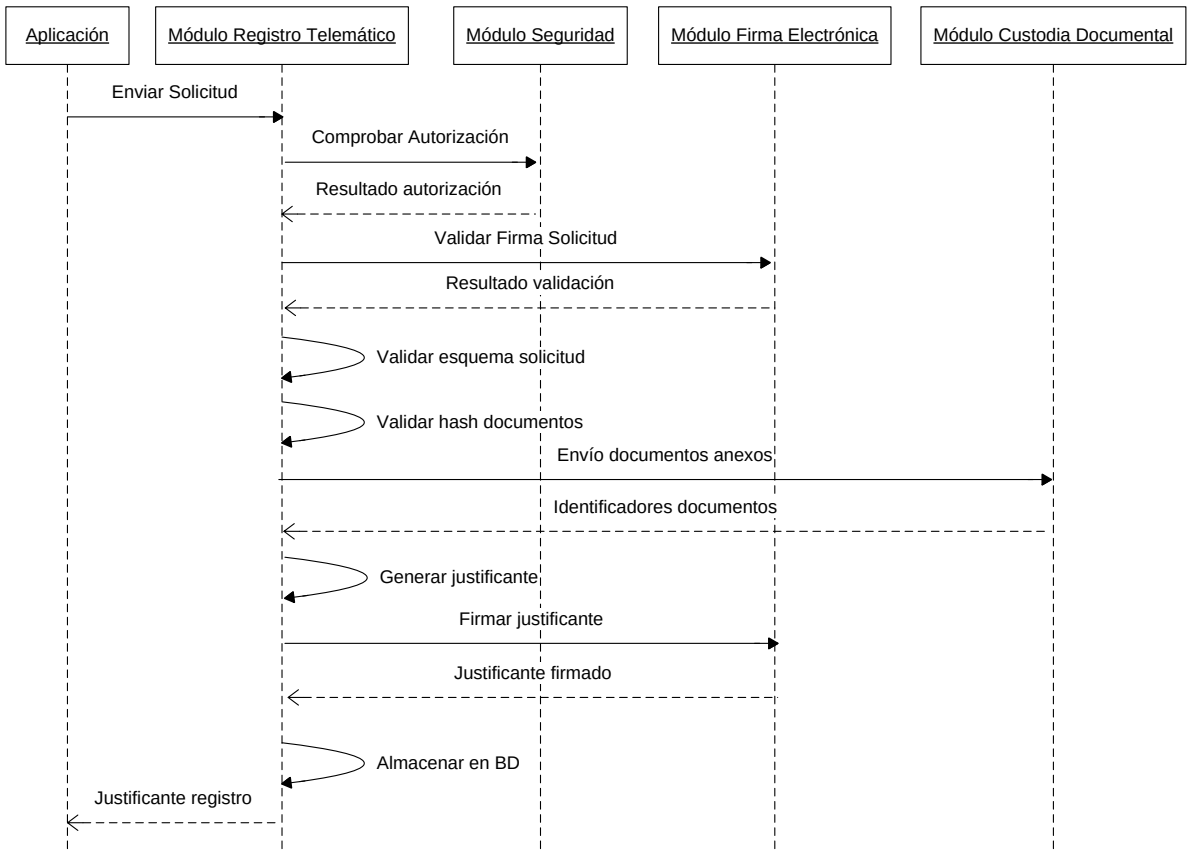

Ilustración 20. Diagrama de secuencia: Envío de una nueva solicitud

4.3.1.3.4.2 Búsqueda de Registros

PARÁMETROS DE ENTRADA		
NOMBRE	TIPO	DESCRIPCIÓN
idAplicacion	String	Tipo: **requerido** Identificador de la aplicación
ticket	String	Tipo: **requerido** Ticket de control para gestionar la seguridad. Será proporcionado a las aplicaciones por el módulo de seguridad
numRegistro	String	Tipo: **opcional** Nº de registro obtenido en el proceso de Envío de nueva solicitud
numExpediente	String	Tipo: **opcional** Número de expediente
idAplicación	String	Tipo: **opcional** ID de aplicación que ha generado la solicitud
nifRemitente	String	Tipo: **opcional** NIF del emisor de la solicitud
codDestino	String	Tipo: **opcional** Código identificador del organismo de destino
codAsunto	String	Tipo: **opcional** Código de asunto
fechaRecepcionDesde	Date	Tipo: **opcional** Intervalo de fechas en las que se recibió la solicitud
fechaRecepcionHasta	Date	

Tabla 33. Proceso de Búsqueda de Registros. Parámetros de Entrada

PARÁMETROS DE SALIDA		
NOMBRE	TIPO	DESCRIPCIÓN
resultado	Bool	Posibles valores: • TRUE en caso de ejecución correcta • FALSE en caso de producirse un error
numsRegistro	String	Listado de números de registro congruentes con los criterios de búsqueda
descripción	String	En caso de que error = FALSE • Vacío En caso de que error = TRUE • Descripción del error

Tabla 34. Proceso de Búsqueda de Registros. Parámetros de Salida

4.3.1.3.4.2.1 Diagrama de secuencia

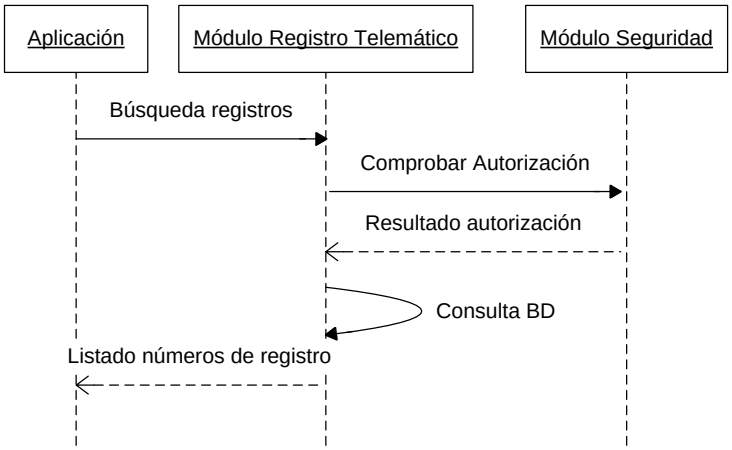

Ilustración 21. Diagrama de secuencia: Búsqueda de registros

4.3.1.3.4.3 Obtener Registro

PARÁMETROS DE ENTRADA		
NOMBRE	TIPO	DESCRIPCIÓN
idAplicacion	String	Tipo: **requerido** Identificador de la aplicación
ticket	String	Tipo: **requerido** Ticket de control para gestionar la seguridad. Será proporcionado a las aplicaciones por el módulo de Seguridad
idRegistro	String	Tipo: **requerido** ID de registro
devuelveDocs	Bool	Posibles valores: • TRUE en caso de que se desee obtener los documentos anexos a una solicitud • FALSE en caso contrario

Tabla 35. Proceso de Obtener Registros. Parámetros de Entrada

PARÁMETROS DE SALIDA		
NOMBRE	TIPO	DESCRIPCIÓN
resultado	Bool	Posibles valores: • TRUE en caso de ejecución correcta • FALSE en caso de producirse un error
descripción	String	En caso de que error = vacío En caso de que error = TRUE • Descripción del error
solicitud	String	Solicitud XML correspondiente al Nº de registro
justificante	String	Justificante XML correspondiente al Nº de registro
anexos	Array	Array de documentos con la siguiente estructura: • código (String) • nombre (String) • contenido (String base 64) • hash (String base 64) • algoritmoHash (String)

Tabla 36. Proceso de Obtener Registros. Parámetros de Salida

4.3.1.3.4.3.1 Diagrama de secuencia

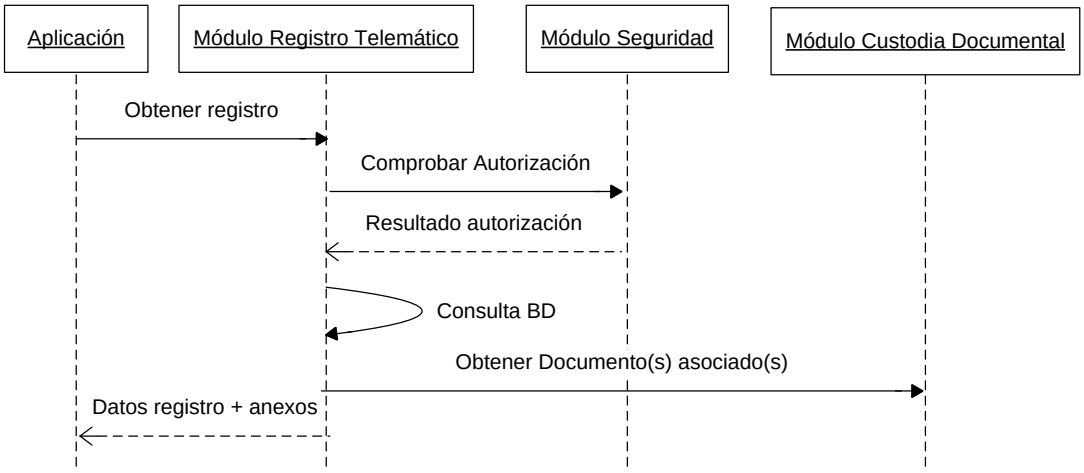

Ilustración 22. Diagrama de Secuencia: Obtener Registro

4.3.1.3.4.4 Eliminar Registro

El servicio de eliminar registro únicamente cambiará el estado del registro a "ELIMINADO".

PARÁMETROS DE ENTRADA		
NOMBRE	TIPO	DESCRIPCIÓN
idAplicacion	String	Tipo: **requerido** Identificador de la aplicación
ticket	String	Tipo: **requerido** Ticket de control para gestionar la seguridad. Será proporcionado a las aplicaciones por el módulo de Seguridad
idRegistro	String	Tipo: **requerido** ID de registro

Tabla 37. Proceso de Eliminar Registros. Parámetros de Entrada

PARÁMETROS DE SALIDA		
NOMBRE	TIPO	DESCRIPCIÓN
resultado	Bool	Posibles valores: • TRUE en caso de ejecución correcta • FALSE en caso de producirse un erro.
descripción	String	En caso de que error = FALSE • Vacío En caso de que error = TRUE • Descripción del error

Tabla 38. Proceso de Eliminar Registros. Parámetros de Salida

4.3.1.3.4.4.1 Diagrama de secuencia

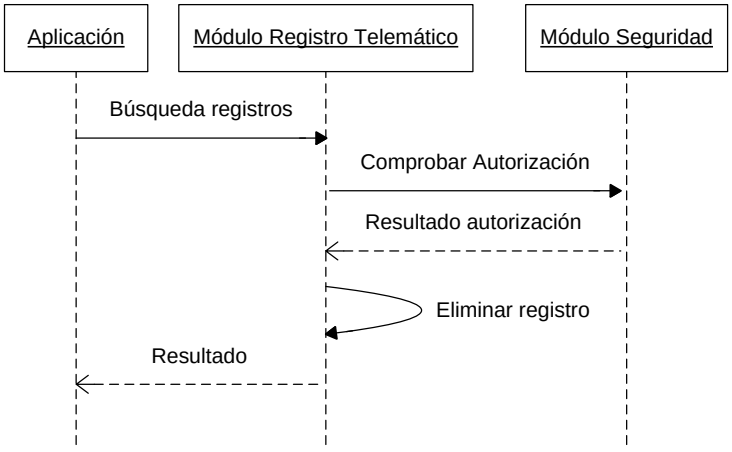

Ilustración 23. Diagrama de Secuencia: Eliminar Registro

4.3.2 Firma Electrónica

Este documento recoge el diseño técnico del módulo de Firma Electrónica.

Está dividido en dos partes principales:

- Interfaz:
  - Se describe de forma detallada cada uno de los servicios que prestará el interfaz.
- Conectores
  - Se describen de forma detallada los conectores que implementan el interfaz definido.

4.3.2.1 Diseño detallado del interfaz

El interfaz del módulo de firma electrónica define los siguientes servicios web, que corresponden a los casos de uso especificados en el modelo:

- Validar Firma
- Validar Certificado
- Obtener Hash
- Firmar
- Firmar secuencial
- Firmar paralelo

4.3.2.1.1 Servicio Validar Firma

El servicio de validación de firmas proporciona un interfaz estándar para validar en servidor la firma de un documento independientemente del formato de firma utilizado.

PARÁMETROS DE ENTRADA		
NOMBRE	TIPO	DESCRIPCIÓN
idAplicacion	String	Identificador de la aplicación. Deberá ser dado de alta en el módulo de Seguridad
formatoFirma	String	El formato de firma admitido (PKCS#7, CMS, XAdES, CADES,…) dependerá de la plataforma de firma electrónica utilizada
datosSinFirmar	String base64	Documento codificado en base 64
firma	String base64	Firma del documento codificada en base 64
ticket	String	Ticket que identifica el emisor con la aplicación

Tabla 39. Servicio de Validar Firma. Parámetros de Entrada

PARÁMETROS DE SALIDA		
NOMBRE	TIPO	DESCRIPCIÓN
resultado	Bool	Posibles valores: • TRUE en caso de ejecución correcta • FALSE en caso de producirse un error
estadoFirma	string	Posibles valores: • Firma válida • Firma no válida • Vacío: en caso de error
descripción	string	En caso de que error = FALSE • Detalle del estado de firma En caso de que error = TRUE • Descripción del error

Tabla 40. Servicio de Validar Firma. Parámetros de Salida

CÓDIGOS DE ERROR	
CÓDIGO	DESCRIPCIÓN
ERR_ACCESO	Aplicación no válida
ERR_CONEX	Error de conexión con plataforma de firma electrónica
ERR_FORMATO	Formato de firma no válido
ERR_FIRMA	Firma no válida

Tabla 41. Servicio de Validar Firma. Códigos de Error

4.3.2.1.2 Servicio Validar Certificado

El servicio de validación de certificados proporciona un interfaz estándar para validar en servidor cualquier certificado X509 v3. Los prestadores de servicios de certificación admitidos dependerán de la plataforma de firma electrónica utilizada.

PARÁMETROS DE ENTRADA		
NOMBRE	TIPO	DESCRIPCIÓN
idAplicacion	String	Identificador de la aplicación. Deberá ser dado de alta en el módulo de Seguridad
certificado	String (base64)	Contenido del certificado codificado en Base 64
ticket	String	Ticket que identifica el emisor con la aplicación

Tabla 42. Servicio de Validar Certificado. Parámetros de Entrada

PARÁMETROS DE SALIDA		
NOMBRE	TIPO	DESCRIPCIÓN
resultado	Bool	Posibles valores: • TRUE en caso de ejecución correcta • FALSE en caso de producirse un error
estadoCertificado	string	Posibles valores: • Válido • Caducado • Revocado • No válido • Vacío: en caso de error
descripción	string	En caso de que error = FALSE • Detalle del certificado En caso de que error = TRUE • Descripción del error

Tabla 43. Servicio de Validar Certificado. Parámetros de Salida

CÓDIGOS DE ERROR	
CÓDIGO	DESCRIPCIÓN
ERR_ACCESO	Aplicación no válida
ERR_CONEX	Error de conexión con plataforma de firma electrónica
ERR_CERTIFICADO	Formato de certificado no reconocido

Tabla 44. Servicio de Validar Certificado. Códigos de Error

4.3.2.1.3 Servicio Obtener Hash

Este servicio proporciona un interfaz estándar para, a partir de un documento, calcular su hash. Los algoritmos empleados para el cálculo del hash dependerán de la plataforma de firma electrónica utilizada.

PARÁMETROS DE ENTRADA		
NOMBRE	TIPO	DESCRIPCIÓN
idAplicacion	String	Identificador de la aplicación. Deberá ser dado de alta en el módulo de Seguridad
datos	String (base64)	Documento del que se desea obtener el hash
algoritmo	String	Algoritmo a emplear para el cálculo del hash
ticket	String	Ticket que identifica el emisor con la aplicación

Tabla 45. Servicio de Obtener Hash. Parámetros de Entrada

PARÁMETROS DE SALIDA		
NOMBRE	TIPO	DESCRIPCIÓN
resultado	Bool	Posibles valores: • TRUE en caso de ejecución correcta • FALSE en caso de producirse un erro
hash	String (base 64)	Hash del documento
descripción	string	En caso de que error = FALSE • Vacío En caso de que error = TRUE • Descripción del error

Tabla 46. Servicio de Obtener Hash. Parámetros de Salida

CÓDIGOS DE ERROR	
CÓDIGO	DESCRIPCIÓN
ERR_ACCESO	Aplicación no válida
ERR_CONEX	Error de conexión con plataforma de firma electrónica
ERR_ALGORITMO	Algoritmo para el cálculo de hash no reconocido

Tabla 47. Servicio de Obtener Hash. Códigos de Error

4.3.2.1.4 Servicio Firmar

Este servicio permite la firma en servidor de un documento. Los formatos de firma disponibles dependerán de la plataforma de firma electrónica utilizada.

PARÁMETROS DE ENTRADA		
NOMBRE	TIPO	DESCRIPCIÓN
idAplicacion	String	Identificador de la aplicación. Deberá ser dado de alta en el módulo de seguridad
formato	String	Formato de firma
nombreDoc	String	Nombre del documento
tipoDoc	String	Tipo de documento
documento	String (base64)	Contenido del documento codificado en base 64
ticket	String	Ticket que identifica el emisor con la aplicación

Tabla 48. Servicio de Firmar. Parámetros de Entrada

PARÁMETROS DE SALIDA		
NOMBRE	TIPO	DESCRIPCIÓN
resultado	Bool	Posibles valores: • TRUE en caso de ejecución correcta • FALSE en caso de producirse un error
descripción	string	En caso de que error = FALSE • Vacío En caso de que error = TRUE • Descripción del error
firma	String (base64)	Firma del documento en base64
idTransaccion	String	Identificador único de la operación en servidor (necesario para posteriores firmas sobre el mismo documento, en paralelo o en secuencial).

Tabla 49. Servicio de Firmar. Parámetros de Salida

CÓDIGOS DE ERROR	
CÓDIGO	DESCRIPCIÓN
ERR_ACCESO	Aplicación no válida
ERR_CONEX	Error de conexión con plataforma de firma electrónica
ERR_FORMATO	Formato de firma no válido
ERR_TIPO	Tipo de documento no válido

Tabla 50. Servicio de Firmar. Códigos de Error

4.3.2.1.5 Servicio Firmar Secuencial

Este servicio permite añadir una nueva firma secuencial a un documento. Los formatos de firma disponibles dependerán de la plataforma de firma electrónica utilizada.

PARÁMETROS DE ENTRADA		
NOMBRE	TIPO	DESCRIPCIÓN
idAplicacion	String	Identificador de la aplicación. Deberá ser dado de alta en el módulo de Seguridad.
idTransaccion	String	Identificador de la transacción en la que se firmó el documento.
ticket	String	Ticket que identifica el emisor con la aplicación

Tabla 51. Servicio de Firmar Secuencial. Parámetros de Entrada

PARÁMETROS DE SALIDA		
NOMBRE	TIPO	DESCRIPCIÓN
resultado	Bool	Posibles valores: • TRUE en caso de ejecución correcta • FALSE en caso de producirse un error
descripción	string	En caso de que error = FALSE • Vacío En caso de que error = TRUE • Descripción del error
firma	String (base64)	Firma del documento en base64
idTransaccion	String	Identificador único de la operación en servidor (necesario para posteriores firmas sobre el mismo documento, en paralelo o en secuencial)

Tabla 52. Servicio de Firmar Secuencial. Parámetros de Salida

CÓDIGOS DE ERROR	
CÓDIGO	DESCRIPCIÓN
ERR_ACCESO	Aplicación no válida
ERR_CONEX	Error de conexión con plataforma de firma electrónica
ERR_TRANSACC	Identificador de transacción no válido

Tabla 53. Servicio de Firmar Secuencial. Códigos de Error

4.3.2.1.6 Servicio Firmar Paralelo

Este servicio permite añadir una nueva firma secuencial a un documento. Los formatos de firma disponibles dependerán de la plataforma de firma electrónica utilizada.

PARÁMETROS DE ENTRADA		
NOMBRE	TIPO	DESCRIPCIÓN
idAplicacion	String	Identificador de la aplicación. Deberá ser dado de alta en el módulo de Seguridad
idTransaccion	String	Identificador de la transacción en la que se firmó el documento
ticket	String	Ticket que identifica el emisor con la aplicación

Tabla 54. Servicio de Firmar Paralelo. Parámetros de Entrada

PARÁMETROS DE SALIDA		
NOMBRE	TIPO	DESCRIPCIÓN
resultado	Bool	Posibles valores: • TRUE en caso de ejecución correcta • FALSE en caso de producirse un error
descripción	string	En caso de que error = FALSE • Vacío En caso de que error = TRUE • Descripción del error
firma	String (base64)	Firma del documento en base64
idTransaccion	String	Identificador único de la operación en servidor (necesario para posteriores firmas sobre el mismo documento, en paralelo o en secuencial)

Tabla 55. Servicio de Firmar Paralelo. Parámetros de Salida

CÓDIGOS DE ERROR	
CÓDIGO	DESCRIPCIÓN
ERR_ACCESO	Aplicación no válida
ERR_CONEX	Error de conexión con plataforma de firma electrónica
ERR_TRANSACC	Identificador de transacción no válido

Tabla 56. Servicio de Firmar Paralelo. Códigos de Error

4.3.2.2 Diseño detallado conector @FIRMA

4.3.2.2.1 Introducción

El Ministerio de Administraciones Públicas ha implantado una plataforma de validación y firma electrónica con el objetivo de facilitar a las aplicaciones los servicios necesarios para implementar la autenticación y firma electrónica avanzada.

4.3.2.2.2 Descripción @Firma

La versión actual de la plataforma @Firma es la v5.0. Está basada en software libre y estándares abiertos y desarrollada en java:

- Servidor web apache
- Servidor de aplicaciones JBoss
- Liberia Axis

4.3.2.2.2.1 Formatos de firma electrónica admitidos

- PKCS#7
- CMS:
  - Proviene del formato PKCS#7
  - Recomendación del IETF, RFC 2630
- XMLdSig
  - Recomendado por el W3C
  - Basado en XML
- XAdES
  - Estandarizado por ETSI TS 101 903
  - Amplía las posibilidades del estándar XMLdSig
- CAdES: CMS Advanced Electronic Signatures
  - Estandarizado por ETSI TS 101 733
  - Incorpora información a la firma CMS

4.3.2.2.2.2 Requisitos técnicos de integración

Para la integración con los servicios web de @Firma es necesario tener en cuenta los siguientes condicionantes:

- Los interfaces de los servicios web proporcionados cumplen con el estándar Basic Profile v1.1 del Web service Interoperability (WS-I) de OASIS.
- Los mensajes se intercambian utilizando XML-SOAP, siendo obligatorio que la petición sea realizada en codificación UTF-8 vía http (puerto 8080) o vía HTTPS.
- Los mensajes de respuesta de @Firma son firmados haciendo uso del certificado público de @Firma. La firma se inserta de acuerdo al Binary Security Token según WSS 1.0 de OASIS.

4.3.2.2.2.3 Implementación de seguridad en servicios web

Los métodos empleados son:

- Sin método de autorización
- Usuario y contraseña: mediante este método es necesario que las peticiones SOAP realizadas contengan el tag de seguridad "UserNameToken" y que el usuario y contraseña contenidos en dicho tag estén registrados en @Firma
- Certificado: Las aplicaciones deben contener el tag "BinarySecurityToken" y deben estar firmadas por alguno de los certificados registrados en @Firma para la aplicación. La parte pública de estos certificados debe estar ubicada en el keystore "AlmacenAutorizacionWS" de @Firma.

4.3.2.2.2.4 Catálogo de servicios

- Servicios de validación
  - Validación de certificados
  - Obtención de información de certificados
  - Validación de firma electrónica de múltiples formatos
  - Validación de sellos de tiempo
  - Validar firma de bloques completo
  - Validar firma de bloques documento
- Servicios de firma electrónica:
  - Firma electrónica simple
  - Firma electrónica en paralelo (CoSign)
  - Firma electrónica en cascada o secuencial (CounterSign)
- Servicios auxiliares y de administración de transacciones
  - Obtener identificadores de documentos de un bloque de firmas generado por @firma 4.0.
  - Obtener identificadores de documentos de un bloque de firmas.
  - Obtener información de un bloque de firmas generado por @firma 4.0.
  - Obtener información de un bloque de firmas.
  - Almacenar hash de documento.
  - Eliminar el contenido de un documento.
  - Obtener identificador de un documento.
  - Obtener el contenido de un documento.
  - Obtener el contenido de un documento haciendo uso de su identificador.
  - Obtener transacciones de firma.
  - Obtener transacciones de firma por fecha.

- Obtener la firma electrónica de una transacción.
- Obtener transacciones de firma por referencia.
- Obtener bloque de firmas.

4.3.2.2.2.5 Certificados reconocidos por @Firma

@Firma admite los certificados de los prestadores que se encuentran inscritos en el registro de la Secretaria de Estado de Telecomunicaciones y para la Sociedad de Información, del Ministerio de Industria, Turismo y Comercio, conforme a lo establecido en el artículo 30 de la Ley 59/2003, de 19 de diciembre, de firma electrónica.

- DNIe (DNI electrónico)
- Camerfirma
- Izenpe
- CATCert (Agencia Catalana de Certificación)
- ANF AC (Sistema abierto de certificación electrónica)
- SCR (Servicio de Certificación de los Registradores)
- ACA (Autoridad de Certificación de la Abogacía)
- ACCV (Autoridad de Certificación de la Comunidad Valenciana)
- ANCERT (Agencia Notarial de Certificación)
- FNMT (Fábrica Nacional de Moneda y Timbre)
- Firmaprofesional
- Banesto CA

4.3.2.2.3 Modelo de datos

El modelo de datos se representa empleando la nomenclatura propia de MySql. El gráfico que corresponde a dicho modelo es el siguiente:

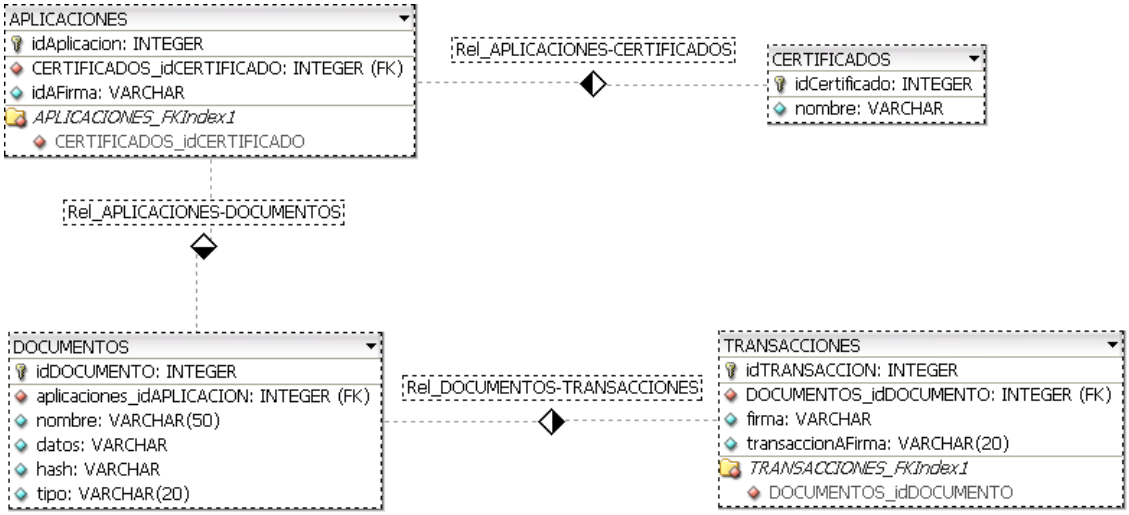

Ilustración 24. Modelo de Datos: Módulo de Firma

TABLA: APLICACIONES		
COLUMNA	TIPO	DESCRIPCIÓN
idAplicacion	Integer	• No nulo • Clave primaria
idFirma	Varchar	• No nulo
CERTIFICADOS_idCERTIFICADO (FK)	Integer	Determina el nombre del certificado que se utilizará en las operaciones de la aplicación

Tabla 57. Modelo de Datos: Aplicaciones

TABLA: DOCUMENTOS		
COLUMNA	TIPO	DESCRIPCIÓN
idDocumento	Integer	• No nulo • Clave primaria
nombre	Varchar	• No nulo
datos	Varchar	Contenido del documento codificado en base64
hash	Varchar	Hash del documento codificado en base64
tipo	Varchar(20)	Tipo de documento
APLICACIONES_idAPLICACION (FK)	Integer	Asocia el documento a una aplicación

Tabla 58. Modelo de Datos: Documentos

TABLA: CERTIFICADOS		
COLUMNA	TIPO	DESCRIPCIÓN
idCertificado	Integer	• No nulo • Clave primaria
nombre	Varchar	• Nombre del certificado dado de alta en @Firma

Tabla 59. Modelo de Datos: Certificados

TABLA: TRANSACCIONES		
COLUMNA	TIPO	DESCRIPCIÓN
idTransaccion	Integer	• No nulo • Clave primaria
firma	Varchar	Contiene la firma asociada a la transacción (en caso de que se haya llevado a cabo correctamente)
transaccionAFirma	Varchar	Contiene el identificador de transacción devuelto por @Firma
DOCUMENTOS_idAPLICACION (FK)	Integer	Asocia la transacción a un documento: todas las transacciones (operaciones de firma en @Firma) están asociadas a un documento

Tabla 60. Modelo de Datos: Transacciones

4.3.2.2.4 Procesos

4.3.2.2.4.1 Validar firma

Parámetros del servicio Validar Firma de @Firma

PARÁMETROS DE ENTRADA		
NOMBRE	TIPO	DESCRIPCIÓN
idAplicacion	string	Tipo: **requerido**. Identificador de la aplicación. Deberá ser dado de alta en el módulo de seguridad
firmaElectrónica	base64Binary	Tipo: **requerido**. Firma electrónica a validar codificado en base64
formatoFirma	string	Tipo: **opcional** Valores posibles: • CMS (Por defecto) • PKCS7 • XADES-BES • XADES-T • XMLDSIG • CADES
hash	base64Binary	Tipo: **opcional** En caso de indicarse, deberá introducirse también el parámetro algoritmoHash
algoritmoHash	string	Tipo: **opcional** Valores posibles: • MD2 • MD5 • SHA • SHA1 • SHA256 • SHA384 • SHA512
datos	base64Binary	Tipo: **requerido**. Datos cuya firma se va a validar codificados en base64

Tabla 61. Proceso Validar Firma: Parámetros de Entrada

El conector simplificará el servicio web Validar Firma de @Firma.

Únicamente ofrecerá la posibilidad de incluir los parámetros requeridos y el parámetro formatoFirma.

El Id de aplicación necesario para @Firma se obtendrá de la tabla Aplicaciones.

PARÁMETROS DE SALIDA		
NOMBRE	TIPO	DESCRIPCIÓN
estado	Bool	Indica si la operación se ha llevado a cabo correctamente Posibles valores: • true • false
proceso	string	Indica si se ha completado el proceso de validación
detalle	string	Indica el resultado de cada una de las etapas de la validación de la firma
conclusión	string	Indica el resultado final del proceso de validación de firma

Tabla 62. Proceso Validar Firma: Parámetros de Salida

4.3.2.2.4.1.1 Diagrama de secuencia

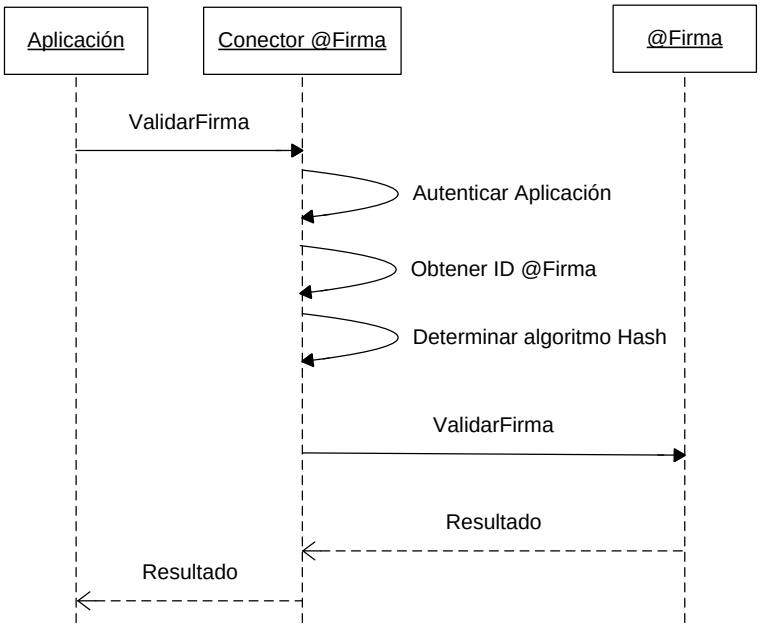

Ilustración 25. Diagrama de Secuencia: Validar Firma

El conector será el encargado de obtener, mediante una consulta a la tabla Aplicaciones, el identificador asociado a la aplicación en la plataforma @Firma. El algoritmo hash a utilizar se parametrizará en un archivo de configuración.

Posteriormente, realizará la llamada al servicio web de @Firma Validar Firma y devolverá el resultado a la aplicación invocante.

4.3.2.2.4.2 Validar Certificado

Parámetros del servicio Validar Certificado de @Firma.

PARÁMETROS DE ENTRADA		
NOMBRE	TIPO	DESCRIPCIÓN
idAplicacion	string	Tipo: **requerido**. Identificador de la aplicación. Deberá ser dado de alta en el módulo de seguridad
certificado	base64Binary	Tipo: **requerido**. Contenido del certificado a validar codificado en base 64
modoValidación	Integer	Tipo: **requerido** Posibles valores: • 0, para una validación simple. Donde se validará la caducidad, integridad y confianza del certificado • 1, para una validación intermedia. Donde se validará la misma información del caso 0 + estado de revocación • 2, para una validación compleja. Donde se validará la misma información del caso 1 + validación de la cadena de confianza al completo El conector implementará por defecto el tipo 1
obtenerInfo	Bool	Tipo: **requerido** Posibles valores: • TRUE: se extraerá información del certificado. • FALSE: no se extraerá información del certificado.

Tabla 63. Proceso Validar Certificado: Parámetros de Entrada

PARÁMETROS DE SALIDA		
NOMBRE	TIPO	DESCRIPCIÓN
infoCertificado	Estructura de datos	Devuelve información del certificado en caso de que el parámetro de entrada obtenerInfo = true
resultadoValidación	Estructura de datos	Elementos: • ValidaciónSimple: resultado de la validación de caducidad, integridad y confianza del certificado. (modoValidación = 0) • ValidaciónEstado resultado de la validación del certificado. (modoValidación = 1) • ValidaciónCadena: resultado de la validación de la cadena de confianza del certificado. (modoValidación = 2)

Tabla 64. Proceso Validar Firma: Parámetros de Salida

4.3.2.2.4.2.1 Diagrama de secuencia

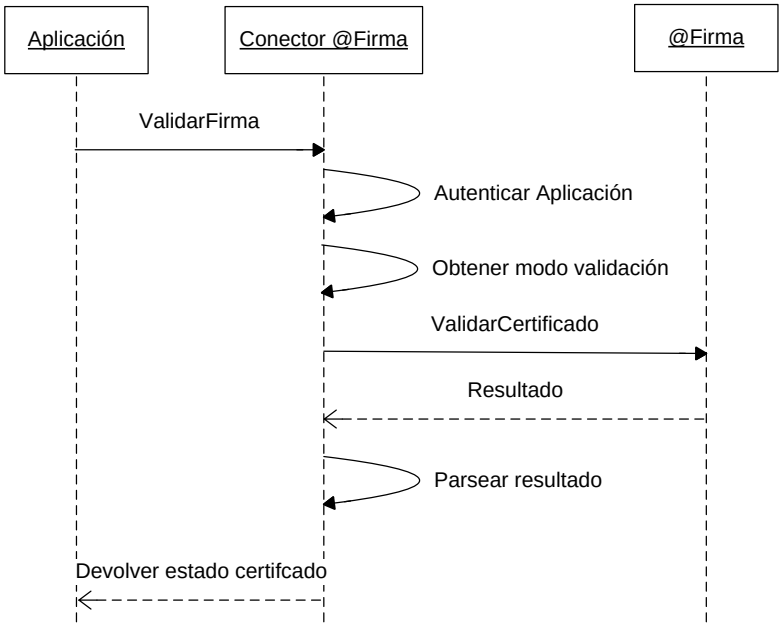

Ilustración 26. Diagrama de Secuencia: Validar Certificado

El conector validará la aplicación y obtendrá el modo de validación de un fichero de configuración. Posteriormente invocará al servicio Validar Certificado de @Firma y comprobará el resultado para devolverle a la aplicación el estado del certificado, que puede ser:

- VALIDO
- CADUCADO
- REVOCADO

4.3.2.2.4.3 Obtener Hash

@Firma no proporciona ningún servicio específico para el cálculo del hash de un conjunto de datos.

Será necesario implementar el código necesario para proporcionar el hash según los siguientes algoritmos:

- MD2
- MD5
- SHA
- SHA1
- SHA256
- SHA384
- SHA512

PARÁMETROS DE ENTRADA		
NOMBRE	TIPO	DESCRIPCIÓN
idAplicacion	string	Tipo: **requerido**. Identificador de la aplicación. Deberá ser dado de alta en el módulo de seguridad
datos	base64Binary	Tipo: **requerido**. Datos de los que se desea obtener el hash codificados en base64
algoritmo	string	Tipo: **opcional**. **Por defecto**: SHA1 Algoritmo deseado para la obtención del hash

Tabla 65. Proceso Obtener Hash: Parámetros de Entrada

PARÁMETROS DE SALIDA		
NOMBRE	TIPO	DESCRIPCIÓN
resultado	Bool	Posibles valores: • TRUE: resultado correcto • FALSE: error
descripción	string	Posibles valores: • Si resultado = TRUE: vacío • Si resultado = FALSE: descripción del error
hash	base64Binary	Hash del documento codificado en base64

Tabla 66. Proceso Obtener Hash: Parámetros de Salida

4.3.2.2.4.4 Firmar

En la invocación al servicio web de @Firma es necesario indicar a la plataforma cuál es el certificado con el que se debe llevar a cabo la operación de firma. Es necesario que, de forma previa, se dé de alta el certificado en la herramienta de Administración de @Firma.

Antes de la llamada al servicio web de firma en servidor es necesario que el documento a firmar haya sido dado de alta en el módulo de Custodia. **El conector realizará ese proceso de forma transparente a las aplicaciones.**

Parámetros del servicio Firma Servidor de @Firma.

PARÁMETROS DE ENTRADA		
NOMBRE	TIPO	DESCRIPCIÓN
idAplicacion	string	Tipo: **requerido**. Identificador de la aplicación. Deberá ser dado de alta en el módulo de seguridad
idDocumento	string	Tipo: **requerido**. Identificador único del documento a firmar. Se deberá haber registrado previamente mediante las interfaces proporcionadas por el módulo de custodia
firmante	string	Tipo: **requerido**. Alias del certificado con el que se desea firmar. Deberá haber sido dado de alta de forma previa a través de la herramienta de administración de @Firma
idReferencia	string	Identificador externo a @Firma. Puede utilizarse para referenciar el documento de forma interna
algoritmoHash	string	Tipo: **opcional** Por defecto**: SHA1**
formatoFirma	string	Formato en el que se desea realizar la firma. Tipo: **opcional** Por defecto**: CMS**

Tabla 67. Proceso Firmar Servidor: Parámetros de Entrada

PARÁMETROS DE SALIDA		
NOMBRE	TIPO	DESCRIPCIÓN
estado	Bool	Posibles valores: • TRUE: resultado correcto • FALSE: error
descripción	string	Descripción del error o del proceso
idTransaccion	string	Identificador único de la transacción generada
firmaElectrónica	base64Binary	Firma electrónica codificada en base 64
formatoFirma	string	Formato de firma utilizado

Tabla 68. Proceso Firmar Servidor: Parámetros de Salida

4.3.2.2.4.4.1 Diagrama de secuencia

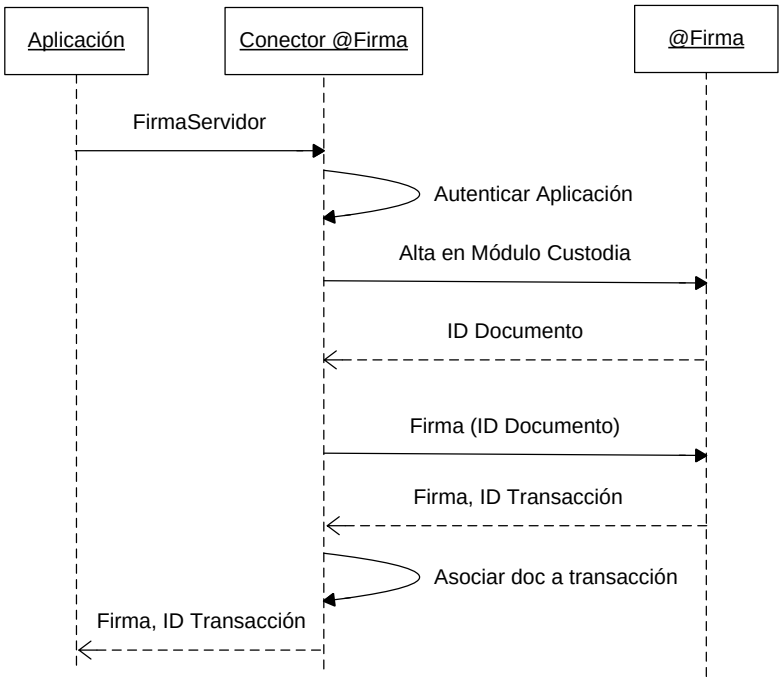

Ilustración 27. Diagrama de Secuencia: Firmar

Descripción del proceso:

- **Autenticar aplicación**: mediante una consulta a la tabla Aplicaciones se comprueba que la aplicación está dada de alta en el sistema y se obtiene el identificador de aplicación para invocar a los servicios web de @Firma.
- **Alta en el módulo de custodia**: de forma transparente para la aplicación cliente, se invoca al servicio *Almacenar Documento* del módulo de custodia y se obtiene el ID de documento de @Firma, necesario para la operación de Firmar Servidor. Posteriormente, se da de alta el documento en la tabla Documentos. (Opcionalmente será posible invocar al servicio web de obtención de hash para almacenar el hash del documento).
- **Invocación al servicio web Firmar Servidor**. Será necesario obtener el alias del certificado mediante una consulta a la tabla Aplicaciones y el algoritmo de hash mediante consulta a fichero de configuración.
- Si el resultado de la invocación al servicio web Firmar Servidor es correcto, se dará de alta un nuevo registro en la tabla transacciones y se devolverá a la aplicación usuaria la firma y el identificador de transacción (necesario para posteriores firmas en paralelo o secuenciales).

4.3.2.2.4.4.2 Alta en el módulo de custodia

De forma previa al proceso de firma es necesario dar de alta el documento en el módulo de custodia. Para ello es necesario invocar al servicio web *Almacenar Documento*. A continuación se describen los parámetros de entrada/salida del servicio *Almacenar Documento*.

PARÁMETROS DE ENTRADA		
NOMBRE	TIPO	DESCRIPCIÓN
idAplicacion	string	Tipo: **requerido** Identificador de la aplicación. Deberá ser dado de alta en el módulo de seguridad
documento	base64Binary	Tipo: **requerido** Contenido del documento cuyos hashes se desean custodiar (codificado en Base64)
nombreDocumento	string	Tipo: **requerido** Nombre del documento a custodiar
tipoDocumento	string	Tipo: **requerido** Tipo del documento a custodiar

Tabla 69. Proceso Firmar: Parámetros de Entrada

PARÁMETROS DE SALIDA		
NOMBRE	TIPO	DESCRIPCIÓN
estado	Bool	Valores posibles: • TRUE: resultado correcto • FALSE: error
descripción	String	Descripción del error o del proceso
idDocumento	String	Identificador único asociado al documento dado de alta en el módulo de custodia

Tabla 70. Proceso Firmar: Parámetros de Salida

4.3.2.2.4.5 Firmar Secuencial

A continuación se describen los parámetros del servicio web *Firma Servidor CounterSign* (Firma secuencial ó multifirma counterSignature).

PARÁMETROS DE ENTRADA		
NOMBRE	TIPO	DESCRIPCIÓN
idAplicacion	string	Tipo: **requerido** Identificador de la aplicación. Deberá ser dado de alta en el módulo de seguridad
idTransaccion	string	Tipo: **requerido** Identificador único de la transacción de firma sobre la que se desea hacer la firma secuencial
firmante	string	Tipo: **requerido** Alias del certificado dado de alta en @Firma
idReferencia	string	Tipo: **opcional** Identificador externo a @firma
algoritmoHash	string	Tipo: **opcional** Algoritmo hash a utilizar Por defecto SHA1
firmanteObjetivo	string	Tipo: **opcional** Certificado X509 codificado en base64 del firmante sobre el que realizar la firma secuencial. En caso de no indicarlo, se hará una firma secuencial sobre todos los firmantes localizados en las hojas del árbol de firmantes

Tabla 71. Proceso Firma Secuencial: Parámetros de Entrada

PARÁMETROS DE SALIDA		
NOMBRE	TIPO	DESCRIPCIÓN
estado	Bool	Valores posibles: • TRUE: resultado correcto • FALSE: error
descripción	string	Descripción del error
idTransaccion	string	Identificador único de la transacción generada
firmaElectrónica	base64Binary	Firma electrónica codificada en base 64
formatoFirma	string	Formato de firma utilizado

Tabla 72. Proceso Firma Secuencial: Parámetros de Salida

4.3.2.2.4.5.1 Diagrama de secuencia

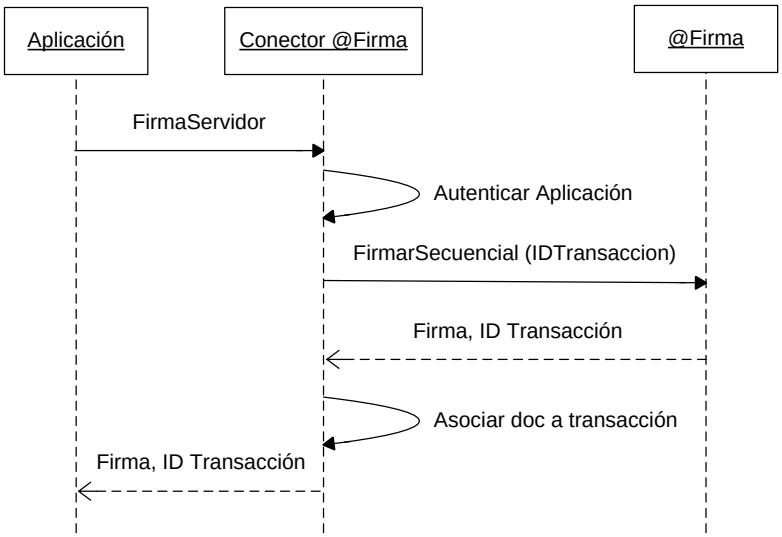

Ilustración 28. Diagrama de Secuencia: Firma Secuencial

Descripción del proceso:

- **Autenticar aplicación**: mediante una consulta a la tabla Aplicaciones se comprueba que la aplicación está dada de alta en el sistema y se obtiene el identificador de aplicación para invocar a los servicios web de @Firma.
- **Invocación del servicio web de firma secuencial** *Firma Servidor CounterSign*. Es necesario proporcionar el identificador de transacción sobre el que se va a realizar la firma secuencial. De forma predeterminada, el conector no introducirá ningún valor en el parámetro de entrada *firmanteObjetivo*.
- Si el resultado de la invocación al servicio web es correcto, se dará de alta un nuevo registro en la tabla Transacciones y se devolverá a la aplicación usuaria la firma y el identificador de transacción (necesario para posteriores firmas en paralelo o secuenciales).

4.3.2.2.4.6 Firmar Paralelo

A continuación se describen los parámetros del servicio web de @firma Firma Servidor CoSignRequest

PARÁMETROS DE ENTRADA		
NOMBRE	TIPO	DESCRIPCIÓN
idAplicacion	string	Tipo: **requerido** Identificador de la aplicación. Deberá ser dado de alta en el módulo de seguridad
idTransaccion	string	Tipo: **requerido** Identificador único de la transacción de firma sobre la que se desea hacer la firma secuencial
firmante	string	Tipo: **requerido** Alias del certificado dado de alta en @Firma
idReferencia	string	Tipo: **opcional** Identificador externo a @firma
algoritmoHash	string	Tipo: **opcional** Por defecto SHA1 Algoritmo hash a utilizar

Tabla 73. Proceso Firma Paralelo: Parámetros de Entrada

PARÁMETROS DE SALIDA		
NOMBRE	TIPO	DESCRIPCIÓN
estado	Bool	Valores posibles: • TRUE: resultado correcto • FALSE: error
descripción	string	Descripción del error
idTransaccion	string	Identificador único de la transacción generada
firmaElectrónica	base64Binary	Firma electrónica codificada en base 64
formatoFirma	string	Formato de firma utilizado

Tabla 74. Proceso Firma Paralelo: Parámetros de Salida

4.3.2.2.4.6.1 Diagrama de secuencia

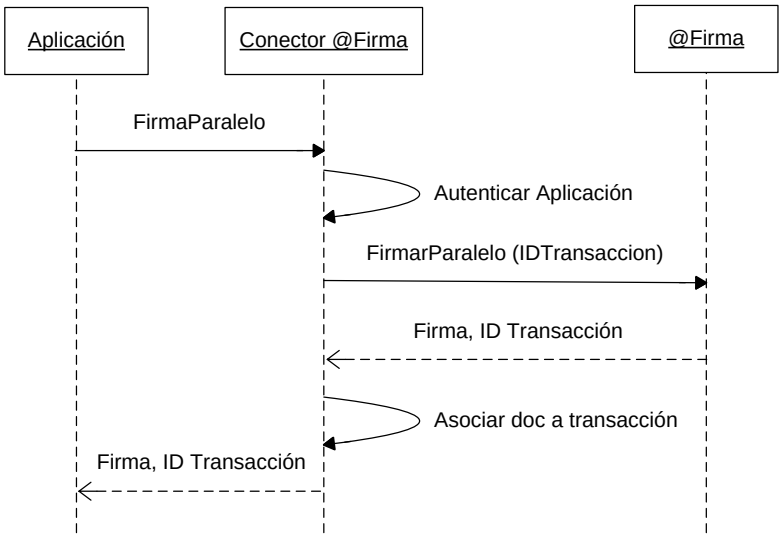

Ilustración 29. Diagrama de secuencia: Firma Paralelo

Descripción del proceso:

- **Autenticar aplicación**: mediante una consulta a la tabla Aplicaciones se comprueba que la aplicación esté dada de alta en el sistema y se obtiene el identificador de aplicación para invocar a los servicios web de @Firma.
- **Invocación del servicio web de firma paralelo** *Firma Servidor CoSignRequest*. Es necesario proporcionar el identificador de transacción sobre el que se va a realizar la firma secuencial. De forma predeterminada, el conector no introducirá ningún valor en el parámetro de entrada *firmanteObjetivo*.
- Si el resultado de la invocación al servicio web es correcto, se **dará de alta un nuevo registro** en la tabla Transacciones y se devolverá a la aplicación usuaria la firma y el identificador de transacción (necesario para posteriores firmas en paralelo o secuenciales).

5. PRESUPUESTO

El coste del proyecto se desdoblará en dos apartados, dedicado el primero de ellos a detallar el coste del proyecto y, en el segundo, se mostrará el presupuesto definitivo que sería ofertado.

5.1 Cálculo del Coste

5.1.1 Gasto Personal

Este proyecto ha sido realizado por una única persona en un periodo de ocho meses, con una dedicación aproximada de 20 horas semanales.

Sin embargo, esta persona ha cumplido con diferentes roles en el mismo proyecto, que por tanto tienen diferentes remuneraciones. Además, cada uno de estos roles tiene una carga de tiempo distinta.

A continuación se detalla el gasto dedicado a remunerar a los hipotéticos empleados. El coste real por hora incluye el IRPF, y el gasto de la seguridad social.

	Coste/ hora	Horas de trabajo	Coste total
Analista	25,45 €	74	1.883,30 €
Diseñador	25,45 €	74	1.883,30 €
Programador	19,09 €	440	8.399,60 €
Gestor Calidad	25,45 €	144	3.664,80 €
		TOTAL:	**15.831,00 €**

Tabla 75. Coste Gasto Personal

5.1.2 Equipos

Para el proyecto es necesario un equipo informático. Se supone que el equipo cuesta 1000€ y que tienen un período de amortización de 48 meses. Además se añade el gasto para ADSL empresarial, del orden de 120€ al mes con mantenimiento y línea de teléfono incluidos.

Coste Equipos	166,67 €
Coste ADSL	960,00 €
TOTAL:	**1.126,67 €**

Tabla 76. Coste Equipos

5.1.3 Otros Gastos

Se dedica 120€ para la compra de material fungible, como puede ser papel, tinta u otros materiales necesarios. Así mismo, también se dedicarán 300€ para viajes y dietas, y sobre el total, se destina un 20% para otros gastos imprevistos.

Material Fungible	120,00 €
Viajes y Dietas	300,00 €
Otros gastos	3.475,53 €

Tabla 77. Coste Otros Gastos

5.1.4 Total

Teniendo en cuenta los anteriores apartados, además de incluir un 5% dedicado al riesgo y un 15% para beneficios, el precio final del proyecto sin IVA será el siguiente:

1. Gasto Personal	15.831,00 €
2. Equipos	1.126,67 €
3. Material Fungible	120,00 €
4. Viajes y Dietas	300,00 €
5. Otros gastos	3.475,53 €
Total	20.853,20 €
Riesgo y Beneficio	4.170,64 €
TOTAL	**25.023,84 €**
TOTAL CON 16% de IVA	**29.027,65 €**

Tabla 78. Coste Total

El coste del proyecto asciende a *veinticinco mil veintitrés con ochenta y cuatro céntimos de euro* (25.023,84€) más el correspondiente importe del IVA, que en el momento de la elaboración de este documento está establecido en el 16%.

5.2 Presupuesto Final

Una vez incluido el IVA, el coste total del proyecto asciende a *veintinueve mil veintisiete, con sesenta y cinco céntimos de euro*.

Presupuesto Final	**29.027,65 €**

6. CONCLUSIONES

Una vez realizado el trabajo, puede concluirse que se han **cumplido satisfactoriamente los objetivos generales** del proyecto. A modo de resumen, estos objetivos son el desarrollo de dos servicios Web que implementen un registro telemático y un servicio de firma digital.

A lo largo de todo el desarrollo del proyecto, el uso de la **arquitectura orientada a servicios ha demostrado ser muy versátil**, ya que se ha podido trabajar en los diferentes módulos individualmente, incluso cuando no estaban terminados y la funcionalidad era incompleta. También ha ofrecido la ventaja de que los cambios realizados en un módulo se reflejan automáticamente en los demás.

Además, los servicios base han podido ser desarrollados y probados a pesar de no encontrarse la aplicación maestra operativa. Por ello, independientemente de que ésta llegue a crearse o no, se ha constatado que son funcionales para aplicaciones de gestión pública.

El uso de esta arquitectura ha propiciado también que la aplicación resultante sea **flexible**, es decir, que es completamente **independiente del procesador y del sistema operativo**, lo que permite que pueda ser utilizada por cualquier administración, independientemente del hardware y software del que se disponga previamente.

En cambio, el desarrollo de una aplicación orientada a servicios tiene como parte negativa el menor rendimiento que puede ofrecer frente a una aplicación desarrollada expresamente para un fin concreto. Sin embargo, dado que en este proyecto el tiempo de respuesta no es un factor crítico, pero la flexibilidad sí lo es, se puede concluir que la **elección de arquitectura ha sido acertada.**

Por otra parte, para el desarrollo de los servicios se han empleado diversas herramientas de libre distribución. Éstas encapsulan funcionalidad específica, como puede ser la comunicación o la persistencia en la base de datos, lo que aumenta la capacidad de adaptarse a diferentes entornos de la herramienta. De esta manera también se comprobó la existencia de un **amplio abanico de opciones para los desarrolladores Java** que facilitan la creación de aplicaciones y mejoran la calidad final del producto.

La contrapartida del uso de este tipo de aplicaciones es la dependencia que puede crearse con otros proveedores de software sin ningún tipo de vínculo contractual, por lo que no son responsables de los errores que sus programas puedan causar. Además, tampoco están obligados

realizar ningún tipo de mantenimiento y no suelen disponer de ayuda técnica para el desarrollador.

También es problemática la depuración en tiempo de desarrollo de este tipo de herramientas de software libre, ya que los errores no están siempre bien detallados y la tarea de solucionarlos puede ser laboriosa y por tanto elevar los costes de mano de obra.

7. LÍNEAS FUTURAS

En este apartado se muestran las directrices a seguir para ampliar y mejorar la funcionalidad del proyecto a corto plazo. Las líneas futuras del proyecto siguen dos vertientes distintas.

En primer lugar, dado que el alcance del proyecto está acotado a dos de los servicios que formarán parte de una aplicación mayor, es posible continuar el proyecto implementando la propia aplicación.

Por otra parte, dado que esta tarea es independiente del proyecto que se ha realizado, puede resultar más interesante perfeccionar y ampliar los propios servicios ofrecidos. En este caso, existen varias posibilidades:

- Podrían incluirse diferentes **bibliotecas de idiomas**. De esta manera, se facilitaría que el registro sea útil para diferentes comunidades autónomas o países
- **Ampliar el catálogo** de **firmas** electrónicas y **certificados** aceptados por el servicio de firma.
- Debido a que los servicios se realiza vía Web, pueden ser objetivo de ataques de piratas informáticos o "hackers" por lo que podría efectuarse una revisión específica en busca de **vulnerabilidades,** que serían **identificadas y subsanadas**.
- La revisión del código, pese a que no se requiere la realización de operaciones excesivamente intensivas, podría suponer una **mejora en la velocidad de ejecución** de las peticiones.

8. BIBLIOGRAFÍA

8.1 Administración electrónica

- España. *Ley 11/2007, de 22 de junio, de acceso electrónico de los ciudadanos a los Servicios Públicos*. *Boletín Oficial del Estado*, 23 de Junio de 2007. Disponible en Web <http://www.boe.es/boe/dias/2007/06/23/pdfs/A27150-27166.pdf>
- España. Ministerio de Economía y Hacienda. *La Administración electrónica y servicio a los ciudadano.* Febrero de 2009. Disponible en web: <http://www.meh.es/Documentacion/Publico/SGT/e-administracion.pdf>
- España. Ministerio de Política Territorial. *Servicios on line*. Disponible en Web: <http://www.map.es/servicios/servicios_on_line.html>
- Universidad de Málaga. *Reglamento del registro telemático*. Disponible en web: http://www.uma.es/secretariageneral/normativa/propia/disposiciones/pro/regl_registrotelematico.htm
- España. Centro de Transferencia de Tecnología. *Servicios públicos digitales*. Disponible en Web:< http://www.ctt.map.es/web/cache/offonce/servicios>

8.2 Registro telemático

- España. Junta de Andalucía. *@RIES*. Disponible en Web <https://ws024.juntadeandalucia.es/pluton/adminelec/ArTec/aries.jsp>
- GADD - Grupo Meana. *SIAC (Sistema de Información y Atención Ciudadana con Registro Telemático)*, Enero de 2008. Disponible en Web: <http://www.gmeana.com/productos/default.aspx?doc=siac>
- Intecna. *OAC (Oficina de Atención al ciudadano)*. Disponible en Web: <http://www.intecna.es/index.php?option=com_content&task=view&id=46&Itemid=116>

8.3 Firma electrónica

- España. *Ley 59/2003, de 19 de diciembre, de firma electrónica*. *Boletín Oficial del Estado*, 19 de Diciembre de 2003. Disponible en Web <http://www.boe.es/boe/dias/2003/12/20/pdfs/A45329-45343.pdf>
- España. Ministerio de Fomento. *Firma Electrónica.* Disponible en Web <http://www.fomento.es/MFOM/LANG_CASTELLANO/OFICINA_VIRTUAL/FIRMA_ELECTRONICA/>
- España. Junta de Andalucía. *@FIRMA*. Disponible en Web: <https://ws024.juntadeandalucia.es/pluton/adminelec/ArTec/afirma.jsp?zona=9&&#general>
- TB-Solutions. *ASF Firma*. Disponible en Web: <http://www.tb-solutions.com/ficheros/tb-solutions_asf-suite_industria.pdf>

8.4 Herramientas de desarrollo

- Oracle - Sun MicroSystems. *Java*. Disponible en Web: <http://java.sun.com/>
- Eclipse Foundation. *Eclipse*. Disponible en Web: <http://www.eclipse.org/>
- Genuitec. *MyEclipse*. Disponible en Web: <http://www.myeclipseide.com/>
- ExoLab Group, Intalio Inc, and Contributors. *The Castor Project*. Disponible en Web: <http://www.castor.org/>
- JUnit.org. *JUnit*. Disponible en Web: <http://www.junit.org/>

8.5 Servicios Web

- W3C. *Web Services Activities*, Octubre de 2009. Disponible en Web:

  <http://www.w3.org/2002/ws>
- CRHISTIANSEN, Eric; CURBERA, Francisco; MEREDITH, Greg; WEERAWARANA, Sanjiva: *Web Services Description Language*, Marzo de 2001. Disponible en Web: <http://www.w3.org/TR/wsdl>
- W3C. *Extensible Markup Language*, Abril de 2009. Disponible en Web: <http://www.w3.org/XML/>

La última consulta a los sitios Webs citados se efectuó el día 10 de Diciembre.